



TUGAS AKHIR - TE145561

RANCANG BANGUN PENGATURAN LAMPU LALU LINTAS BERBASIS MIKROKONTROLER YANG DIKENDALIKAN DENGAN APLIKASI DI ANDROID SECARA NIRKABEL

Fidya Eka Prahesti
NRP 2212039010
Khalid Oki Wibowo
NRP 2212039033

Dosen Pembimbing I
Suwito,ST.,MT.
Dosen Pembimbing II
Eko Pujiyatno Matni, S.Pd

PROGRAM D3 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015



FINAL PROJECT - TE145561

***DESIGN BUILD OF TRAFFIC LIGHTS CONTROLLED BASED ON
MICROCONTROLLED WITH APPLICATIONS IN THE ANDROID
WIRELESSLY***

Fidya Eka Prahesti
NRP 2212039010
Khalid Oki Wibowo
NRP 2212039033

Advisor I
Suwito, ST., MT.
Advisor II
Eko Pujiyatno Matni, S.Pd

*Electrical Engineering D3 Program
Industrial Technology Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2015*

**RANCANG BANGUN PENGATURAN LAMPU LALU LINTAS
BERBASIS MIKROKONTROLER YANG DIKENDALIKAN
DENGAN APLIKASI DI ANDROID SECARA NIRKABEL**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Syarat
Memperoleh Gelar Ahli Madya Teknik
Pada
Bidang Studi Elektro Industri
Program Studi D3 Teknik Elektro
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Menyetujui

Dosen Pembimbing I Dosen Pembimbing II



Suwito, ST.,MT. Eko Pujiyatno Matni, S.Pd
NIP. 19810105 20050 1 004 NIP. 19710330 199403 1 002

**SURABAYA
JUNI, 2015**

RANCANG BANGUN PENGATURAN LAMPU LALU LINTAS BERBASIS MIKROKONTROLER YANG DIKENDALIKAN DENGAN APLIKASI DI ANDROID SECARA NIRKABEL

Nama Mahasiswa 1 : Fidya Eka Prahesti
NRP : 2212 039 010
Nama Mahasiswa 2 : Khalid Oki Wibiwo
NRP : 2212 039 033
Dosen Pembimbing 1 : Suwito, ST., MT.
NIP : 19810105 20050 1 004
Dosen Pembimbing 2 : Eko Pujiyatno Matni.S.pd
NIP : 19710330 199403 1 002

ABSTRAK

Mobilitas kendaraan di jalan raya merupakan proses yang membutuhkan sebuah sistem kontrol pergerakan arus lalu lintas. Waktu tunda yang dimiliki sistem *traffic light* yang ada saat ini hanya mengontrol lampu berdasarkan *timer set*. Sehingga lampu lalu lintas tetap bekerja sesuai *time schedule*. Hal ini menyebabkan kurang efektif. Oleh karena itu dibuatlah sebuah sistem pengaturan lampu lalu lintas dimana lama waktu pada lampu lalu lintas bisa diatur menggunakan android yang dikendalikan oleh polisi lalu lintas secara nirkabel, dengan tujuan untuk mempermudah kerja polisi lalu lintas dalam mengatur kepadatan di ruas jalan *traffic light*.

Prinsip kerja sistem ini adalah saat terjadi peningkatan volume kendaraan di lampu lalu lintas maka polisi lalu lintas cukup merubah waktu pada lampu menggunakan aplikasi di handphone android. Alat ini menggunakan mikrokontroler arduino uno R3, dengan masukan *power supply* 9 volt dan 5 volt. Sedangkan keluarannya terhubung ke rangkaian koneksi *Bluetooth* HC05 dan rangkaian *driver relay* untuk mengaktifkan lampu lalu lintas. Program lampu lalu lintas dikendalikan dengan aplikasi di *handphone* android menggunakan *app inventor*. Jarak komunikasi *Bluetooth* yang dapat dilakukan untuk mengirim dan menerima data adalah sejauh 12 ± 1 meter. RTC yang digunakan dapat menjadi acuan untuk *time shedule* lampu lalu lintas. *Error* yang dihasil dari tegangan keluaran *Power Supply* 5volt sebesar 0,10 - 0,12 volt dan pada *power supply* 9 volt sebesar 0 volt. Penambahan delay waktu dengan app inventor baik satuan maupun puluhan dapat diterima dengan baik dengan presentase *error* sebesar 0% .

Kata kunci : lampu lalu lintas, Arduino, App Inventor, *Relay*, *Bluetooth*, Android, dan RTC.

**DESIGN BUILD OF TRAFFIC LIGHTS CONTROLLED BASED
ON MICROCONTROLLER WITH APPLICATIONS IN THE
ANDROID WIRELESSLY**

Student	1	: Fidy Eka Prahesti
NRP		: 2212 039 010
Student	2	: Khalid Oki Wibiwo
NRP		: 2212 039 033
Advisor	1	: Suwito, ST., MT.
NIP		: 19810105 20050 1 004
Advisor	2	: Eko Pujiyatno Matni, S.pd
NIP		: 19710330 199403 1 002

ABSTRACT

Mobility vehicle on the highway is a process that requires a control system traffic movement. The time delay is owned by the existing traffic light system is currently only controls the lights based on a timer set. So the traffic lights still work appropriate time schedule. This leads to less effective. Therefore made a traffic light control system in which a long time at traffic lights can be set using the android controlled by traffic police wirelessly, with the aim to facilitate the work of the traffic police in regulating kepadan in road traffic light.

The working principle of this system is when there is an increase in the volume of vehicles at traffic lights, the traffic police enough time to change the lamp used in cell phone applications android. This tool uses a microcontroller arduino uno R3, the input power supply 9 volt and 5 volt. While the output is connected to a Bluetooth connection HC05 series and relay driver circuit to activate the traffic light. Program traffic lights are controlled by the application in android phone using the app inventor. Bluetooth communication distance that can be done to send and receive data as far as 12 ± 1 meter. RTC which used to be a reference to time shedule traffic lights. Error to result from the voltage remove the Power Supply 5volt of 0.10 to 0.12 volts and the power supply 9 volts at 0 volts. The addition of delay time with app inventor both units as well as dozens of well received with a percentage error of 0%.

Key Word : Traffic Light, Arduino, App Inventor, Relay, Bluetooth, Android.

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan Syukur dipanjatkan kepada Tuhan Yang Maha Esa yang telah memberikan rahmat dan karunianya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul :

“Rancang Bangun Pengaturan Lampu Lalu Lintas Berbasis Mikrokontroler Yang Dikendalikan Dengan Aplikasi Di Android Secara Nirkabel”

Dan tak lupa saya mengucapkan banyak terima kasih kepada Dosen Pembimbing Tugas Akhir, telah bersedia membimbing dan membantu kami saat pelaksanaan Tugas Akhir. Dengan dilaksanakannya Tugas Akhir ini, mempunyai harapan dapat menerapkan pengalaman dan ilmu yang telah didapatkan pada jenjang yang lebih tinggi. Penyusunan tugas akhir ini penyusun tidak terlepas dari bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis hendak menyampaikan rasa terima kasih kepada pihak – pihak yang telah membantu. Ucapan terima kasih ini kami sampaikan kepada :

1. Allah SWT yang telah memberi rahmatNya dalam pembuatan tugas akhir ini.
2. Kedua orang tua yang telah memberikan dukungan penuh dan teladan bagi penulis.
3. Bapak Suwito,ST.,MT selaku Dosen Pembimbing yang telah meluangkan waktunya untuk membimbing penulis dalam pembuatan tugas akhir ini.
4. Bapak Eko Pujiyatno Matni, S.Pd selaku Dosen Pembimbing selama proses pembuatan tugas akhir ini.
5. Keluarga D3 Tenknik Elektro 2012 atas semangat dan kerjasamanya.,Keluarga D3 Elektro Industri angkatan 2012 kerjasama Disnakertransduk yang selalu ada untu kami.

Akhir kata semoga buku ini dapat bermanfaat bagi kita semua.

Surabaya, Juni 2015

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

JUDUL	i
PENGESAHAN	iii
ABSTRAK	v
<i>ABSTRACT</i>	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
 BAB I PENDAHULUAN	 1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Maksud dan Tujuan.....	2
1.5 Metodologi Penelitian.....	2
1.6 Sitematika Laporan.....	3
1.7 Relevansi	4
 BAB II TEORI PENUNJANG	 5
2.1 <i>Traffic Light</i>	5
2.2 <i>App Inventor</i>	7
2.3 Android.....	9
2.4 <i>Serial Communication</i>	10
2.5 <i>Bluetooth</i>	11
2.6 Mikrokontroller	12
2.7 <i>Driver Relay</i>	14
2.8 <i>Power Supply</i>	15
2.9 (RTC) <i>Real Time Clock</i>	16
 BAB III PERANCANGAN DAN PEMBUATAN	 16
3.1 Perancangan Mekanik.....	18
3.1.1 Layout Jalan Simpang Empat	18
3.1.2 Kotak Penyimpanan Alat Elektronik	19
3.1.3 Kotak Penempatan Lampu Dan Tiang Penyangga	20
3.2 Perancangan Perangkat Keras (<i>Hardware</i>).....	21
3.2.1 Perancangan <i>Power Supply</i>	21
3.2.2 Perancangan Mikrokontroler.....	23

3.2.3 Perancangan <i>Driver Relay</i>	24
3.2.4 Perancangan Komunikasi Serial <i>Bluetooth</i> HCO5	25
3.2.5 Perancangan RTC DS1307	27
3.3 Perancangan Perangkat Lunak (<i>Software</i>)	29
3.3.1 Perancangan Program <i>Traffic Ligh</i>	29
3.3.2 Perancangan Perangkat Lunak Menggunakan Google App Inventor	31
BAB IV PENGUJIAN DAN PENGUKURAN	36
4.1 Pengujian Perangkat Keras (<i>Hardware</i>)	38
4.1.1 Pengujian Pada Input/Ourput Arduino Uno	38
4.1.2 Pengujian Rangkaian <i>Power Supply</i>	39
4.1.3 Pengujian Rangkaian <i>Driver Relay</i>	42
4.1.4 Pengujian Rangkaian Bluetooth HC-05	43
4.1.5 Pengujian RTC (<i>Real Time Clock</i>)	45
4.2 Pengujian Perangkat Lunak (<i>Software</i>).....	46
4.2.1 Pengujian <i>Serial Monitor</i> Untuk Menambah Waktu Lampu Lalu Lintas	46
4.2.2 Pengujian Program MIT App Inventor	49
4.3 Pengujian Alat Secara Keseluruhan.....	51
BAB V PENUTUP	53
5.1 Kesimpulan.....	53
5.2 Saran.....	53
DAFTAR PUSTAKA	55
LAMPIRAN 1 <i>Listing</i> Program	A-1
LAMPIRAN 2 <i>Datasheet</i>	B-1
RIWAYAT HIDUP PENULIS	C-1

DAFTAR TABEL

Tabel 2.1	Produksi Hanphone Android tahun 2015.....	9
Tabel 3.1	<i>Time Schedule</i> Simpang Empat Jl Semolowaru	27
Tabel 3.2	Jadwal Jam Pada <i>Plan</i>	28
Tabel 4.1	Pengujian I/O Ardiono Uno.....	38
Tabel 4.2	Pengujian Rangkaian <i>Power Suply 5V</i>	39
Tabel 4.3	Pengujian Rangkaian <i>Power Suply 9V</i>	41
Tabel 4.4	Pengujian Rangkaian <i>Relay</i>	43
Tabel 4.5	Pengujian Jarak Koneksi Bluetooth	44
Tabel 4.6	Waktu Pengiriman Data Bluetooth.....	44
Tabel 4.7	Pengujian RTC	45
Tabel 4.8	Pengujian Dari Progam App Inventor pada lampu 1 dengan delay awal 10 detik	49
Tabel 4.9	Pengujian Dari Progam App Inventor pada lampu 2 dengan delay awal 10 detik	50
Tabel 4.10	Pengujian Dari Progam App Inventor pada lampu 3 dengan delay awal 10 detik	50
Tabel 4.11	Pengujian Dari Progam App Inventor pada lampu 4 dengan delay awal 10 detik	51

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1. Latar Belakang

Mobilitas kendaraan di jalan raya merupakan proses yang membutuhkan sebuah sistem penyettingan lampu lalu lintas untuk mengatur pergerakan arus lalu lintas. Sistem kendali arus lalu lintas yang telah ada dan dikenal saat ini adalah *traffic light system*. *Traffic light* atau lampu lalu lintas bekerja dengan menggunakan settingan waktu untuk mengendalikan nyala lampu merah, kuning, dan hijau. Dengan waktu tunda yang dimiliki, sistem *traffic light* yang ada saat ini di semolowaru, surabaya mengatur lampu jalan berdasarkan timer set, sehingga bagaimanapun kondisi kendaraan di jalan baik padat maupun senggang lampu lalu lintas tetap bekerja sesuai settingan waktu. Hal ini menyebabkan kurang efektifnya sistem ini pada suatu kondisi, dimana pada saat sebuah ruas jalan senggang, nyala lampu hijaunya begitu lama, sedangkan disaat sebuah ruas jalan begitu padat, nyala lampu merahnya begitu lama.

Sistem *traffic light* yang ada saat ini dirasa membutuhkan pengembangan, sebab jumlah kendaraan di jalan semakin bertambah, sehingga membutuhkan suatu pengaturan jalan yang lebih efektif dan mampu mengontrol kepadatan kendaraan di semolowaru. Oleh karena itu polisi lalu lintas akan mengendalikan dan mengatur lampu lalu lintas secara langsung di lapangan melalui gerakan tangan atau peluit. Meskipun sudah diatur di dalam Undang -Undang Lalu Lintas, solusi ini masih kurang efisien karena menciptakan kebingungan pengguna jalan raya yang disebabkan antara nyala lampu lalu lintas dengan tanda yang diberikan polisi yang bertugas sering berbeda. Hal ini sering memicu terjadinya kecelakaan.

Berdasarkan permasalahan tersebut, maka dibuatlah sebuah sistem pengaturan *traffic light*, yang mana lama waktu pada lampu lalu lintas bisa diatur menggunakan android yang dikendalikan oleh polisi lalu lintas. Lama waktu lampu lalu lintas juga perlu diperhatikan maka perlu adanya tampilan antarmuka pada aplikasi sehingga lama waktu lampu lalu lintas yang diatur dapat dimonitoring. Cara kerja sistem ini, sederhananya adalah saat terjadi peningkatan *volume* kendaraan di *traffic light* maka polisi lalu lintas cukup merubah waktu pada lampu menggunakan aplikasi pada *handphone* android yang sudah dilengkapi program untuk mengaturnya.

1.2. Perumusan Masalah

Melihat latar belakang di atas, maka dirumuskan permasalahan antara lain :

- 1.2.1. Belum adanya sistem pengaturan lampu lalu lintas di jalan semolowaru, surabaya secara nirkabel.
- 1.2.2. Belum adanya pengaturan waktu penyalaaan lampu lalu lintas yang dapat disesuaikan dengan kepadatan di ruas jalan semolowaru, surabaya.
- 1.2.3. Sistem monitoring nyala lampu lalu lintas yang ada pada saat ini belum secara nirkabel di beberapa ruas jalan salah satunya di jalan semolowaru, surabaya.

1.3 Batasan Masalah

Batasan masalah yang akan dibahas dalam tugas akhir ini adalah :

- 1.3.1 Jumlah ruas jalan yang digunakan untuk dikontrol hanyalah persimpangan empat.
- 1.3.2 Pengujian sistem *traffic light* yang ditampilkan di aplikasi android hanya penambahan *delay* lampu hijau.
- 1.3.3 Koneksi antara arduino dengan aplikasi di android menggunakan komunikasi *bluetooth* HC-05.
- 1.3.4 *Traffic light* hanya bisa dikontrol dengan bantuan operator atau polisi lalu lintas yang ada.

1.4 Maksud dan Tujuan

Maksud dan Tujuan dari tugas akhir ini adalah :

- 1.4.1 Dengan menggunakan sistem pengaturan lalu lintas berbasis mikrokontroler secara nirkabel akan menjadikan sistem lalu lintas yang lebih efektif.
- 1.4.2 Membuat sistem pengaturan *traffic light* yang pengoperasiannya berdasarkan waktu yang dapat diatur oleh polisi lalu lintas.
- 1.4.3 Memodifikasi sistem *traffic light* yang ada di jalan semolowaru, surabaya dengan menggunakan aplikasi android .
- 1.4.4 Sebagai syarat kelulusan Tugas Akhir.

1.5 Metodologi Penelitian

Dalam pelaksanaan Tugas Akhir ini ada beberapa tahapan dan metodologi kegiatan yang diantaranya adalah sebagai berikut:

- 1.5.1. Studi Pustaka dan Survey Data Awal

Literatur yang terkait dapat dicari melalui internet, perpustakaan dan survey lapangan tentang sistem *traffic light* yang ada pada saat ini. Pencarian literatur yang terkait dapat berupa komunikasi data melalui wireless, pengukuran besaran arus, pembuatan Interfacing android dan kondisi areal fisik.

1.5.2. Perencanaan dan Pembuatan Alat

Tahapan ini dilakukan dengan perancangan miniatur sistem lalu lintas seperti setting waktu, jumlah ruas jalan, letak *traffic light*, modul *wireless*, modul sensor arus, rangkaian *power supply*.

1.5.3. Perencanaan dan Pembuatan *Software*

Agar hasil rancangan dan pembuatan alat bisa bekerja dengan normal sesuai tujuan Tugas Akhir ini, dibuat perancangan meliputi pembuatan *Interfacing* di Android, algoritma kontrol sensor dengan arduino, arduino dengan bluetooth.

1.5.4. Uji Coba dan Analisis Data

Uji coba dilakukan di Laboratorium elektronika terapan AA104 dengan beberapa peralatan penunjang seperti Clampmeter, Oscilloscope, Multimeter.

1.5.5. Penyusunan Laporan

Penyusunan Laporan yang berisikan informasi tentang sistem lalu lintas yang dibuat, perangkat elektronik, data hasil uji coba yang sudah mencukupi.

1.6 Sistematika Laporan

Dalam penyusunan buku tugas akhir ini, pembahasan mengenai system alat yang dibuat disusun dengan sistematika sebagai berikut :

- 1.6.1 BAB I PENDAHULUAN : membahas tentang latar belakang, permasalahan, tujuan, batasan masalah, sistematika penulisan serta relevansi yang digunakan dalam tugas akhir yang dibuat.
- 1.6.2 BAB II TEORI PENUNJANG : menjelaskan dasar teori yang berisi tentang konsep yang dijadikan landasan dan mendukung dalam perencanaan serta pembuatan alat yang dibuat.
- 1.6.3 BAB III PERANCANGAN SISTEM : dalam bab ini membahas tentang perencanaan dan pembuatan perangkat

keras (*hardware*) yang terdiri atas rangkaian elektronika, desain mekanik serta perangkat lunak (*software*) yang terdiri atas program yang akan digunakan untuk menjalankan alat tersebut.

- 1.6.4 BAB IV PENGUJIAN DAN ANALISA SISTEM : membahas tentang pengujian alat dan analisa data yang didapat dalam pengujian alat.
- 1.6.5 BAB V PENUTUP : berisi tentang kesimpulan alat dari tugas akhir ini dan saran-saran untuk pengembangan alat ini selanjutnya.

1.7 Relevansi

Manfaat/Relevansi dari pembuatan Tugas Akhir ini adalah

- 1.7.1 Memudahkan kerja polisi lalu lintas mengatur kepadatan *volume* kendaraan di persimpangan *traffic light*.
- 1.7.2 Membuat sistem lalu lintas di surabaya menjadi lebih aman dan terhindar dari kecelakaan lalu lintas.
- 1.7.3 Mengatur waktu *traffic light* sesuai dengan kondisi yang terjadi secara langsung di lokasi.
- 1.7.4 Membuat tampilan *interface* dengan aplikasi di android, untuk mengontrol lama waktu lampu lalu lintas.

BAB I

TEORI PENUNJANG

Dalam pembuatan alat pada tugas akhir ini terdapat beberapa perangkat yang digunakan, antara lain rangkaian *power supply*, rangkaian *driver relay*, rangkaian komunikasi bluetooth, dan program *app inventor google*

2.1 Traffic light

Lampu lalu lintas (menurut UU no. 22/2009 tentang Lalu lintas dan Angkutan Jalan merupakan alat pemberi isyarat lalu lintas atau APILL). Lampu lalu lintas adalah lampu yang mengendalikan arus lalu lintas yang terpasang di persimpangan jalan, tempat penyeberangan pejalan kaki (*zebra cross*), dan tempat arus lalu lintas lainnya. Lampu ini yang menandakan kapan kendaraan harus berjalan dan berhenti secara bergantian dari berbagai arah. Pengaturan lalu lintas di persimpangan jalan dimaksudkan untuk mengatur pergerakan kendaraan pada masing-masing kelompok pergerakan kendaraan agar dapat bergerak secara bergantian sehingga tidak saling mengganggu antar-arus yang ada. Lampu lalu lintas telah diadopsi di hampir semua kota di dunia ini. Lampu ini menggunakan warna yang diakui secara universal. Untuk menandakan berhenti adalah warna merah, hati-hati yang ditandai dengan warna kuning, dan hijau yang berarti dapat berjalan. Pada Gambar 2.1 merupakan contoh lampu lalu lintas di jalan.



Gambar 2.1 Lampu Lalu Lintas

Warna yang paling umum digunakan untuk lampu lalu lintas adalah merah, kuning, dan hijau. Merah menandakan berhenti atau sebuah tanda bahaya, kuning menandakan hati-hati, dan hijau menandakan boleh memulai berjalan dengan hati-hati. Biasanya, lampu warna merah mengandung beberapa corak berwarna jingga, dan lampu hijau mengandung beberapa warna biru. Ini dimaksudkan agar orang-orang yang buta warna merah dan hijau dapat mengerti sinyal lampu yang menyala. Di Amerika Serikat, lampu lalu lintas memiliki pinggiran berwarna putih yang dapat menyala dalam kegelapan. Ini bertujuan agar orang yang mengidap buta warna dapat membedakan mana lampu kendaraan dan yang mana lampu lalu lintas dengan posisinya yang vertikal. Lampu lalu lintas pada umumnya dioperasikan dengan menggunakan tenaga listrik. Namun, saat ini sudah perkembangan teknologi lampu lalu lintas dengan tenaga matahari.

Rangkaian *traffic light* ini merupakan rangkaian utama dalam perancangan tugas akhir kami. Komponen-komponen yang ada dalam rangkaian ini adalah pilot lamp. Pilot lamp ini berfungsi sebagai *output* lampu lalu lintas dari arduino. Pilot lamp dihubungkan ke relay dengan tegangan 220V. Pada Gambar 2.2 merupakan bentuk fisik *pilot lamp*.^[1]



Gambar 2.2 Bentuk fisik *pilot lamp*

Traffic light yang kami gunakan mengadopsi dari persimpangan 4 lalu lintas yang ada di Semolowaru Surabaya. Cara kerja sistem traffic light pada kondisi normal ialah arah perpindahan lampu

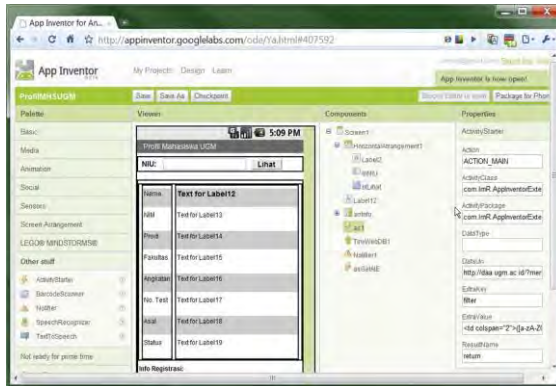
¹**Lampu Lalu Lintas** , URL: http://id.wikipedia.org/wiki/Lampu_lalu_lintas
31 Mei 2015

lalu lintas diatur dengan berlawanan arah jaru jam (counter clockwise). Di Indonesia alur penyalaaan lampu lalu lintas menggunakam sistem berlawanan arah jarum jam agar menghindari kecelakaan antara pengendara bermotor pada saat detik terakhir lampu hijau menyala berganti menjadi hijau di ruas jalan selanjutnya. Lampu lalu lintas bekerja secara bergantian pada tiap jalur sesuai dengan waktu yang sudah ditentukan dengan urutan menyala lampu hijau, lampu kuning dan lampu merah. Pada setiap lampu lalu lintas ,menpunyai nilai *time schedule* yang berbeda. Di setiap jamnya waktu dari lampu lalu lintas di setting berdasakan plan yang sudah dibuat. Nyala lampu secara bergantian, saat salah satu lampu hijau menyala maka lampu yang lainnya menyalah merah. Pergantian lampu merah ke hijau ditandai dengan nyala lampu kuning yang menyala semala 1,5 detik terlebih dahulu.

2.2 *App inventor*²

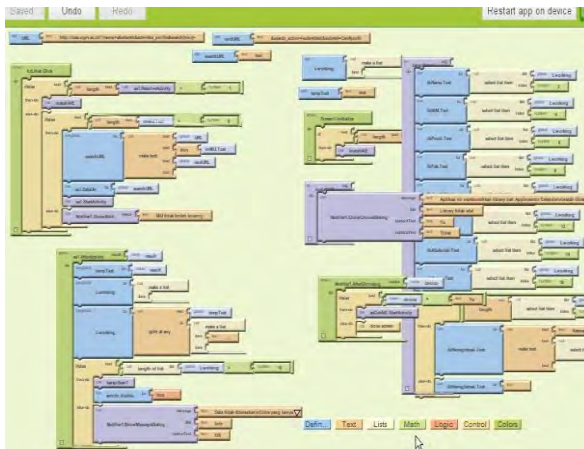
Google app inventor adalah sebuah aplikasi yang disediakan oleh google yang memungkinkan orang untuk membuat aplikasi perangkat lunak untuk os android. *App Inventor* menggunakan *graphical interface* yang memungkinkan *user* melakukan *drag and drop* objek visual untuk membuat aplikasi yang berjalan pada sistem android. Pengembangan aplikasi ini adalah bagaimana kode aplikasi disusun. Berbeda dengan tool pengembangan aplikasi yg lain seperti Eclipse, netbeans, Visual C, Delphi, dll. *Tool* ini tidak menggunakan koding tekstual, melainkan dengan menyusun blok- blok kode. *App Inventor* memiliki dua bagian utama yaitu *App Inventor Designer* dan *Block Editor*. *App Inventor Designer* adalah komponen desainer berjalan pada browser yang digunakan untuk memilih komponen yang dibutuhkan dan mengatur *property* nya. Pada komponen desainer sendiri terdapat 5 bagian, yaitu *palette*, *viewer*, *component*, *media* dan *properties*. Pada Gambar 2.3 dan 2.4 adalah Gambar tampilan dari *App Inventor Designer* dan *Block Editor*.

² Hardesty, Larry 2010. *The MIT roots of Google's new software*, Emereo Publishing, Hal 33.



Gambar 2.3 App Inventor Designer

Block editor adalah bagian dari *App Inventor* untuk merancang logika pemrograman komponen-komponen yang telah dibuat sebelumnya di *App Inventor Designer*. Pada *Block Editor*, *user* dapat menggabungkan blok-blok kode yang terkait. Blok-blok tersebut terdapat pada *drawers* bagian sebelah kiri dari *user interface Block Editor*. *User* hanya perlu *drag* blok yang berada pada *drawers* ke dalam *workspace* dalam rangka untuk menambahkannya ke proyek.



Gambar 2.4 Block Editor

2.3 Android³

Android adalah sistem operasi yang dikembangkan dan dipopulerkan oleh google. Pada awalnya perkembangan OS android ini kurang begitu bagus, hal ini membuat google turun tangan dengan turut melahirkan ponsel google android pertama yang legendaris yang dikenal dengan nama Nexus One. Di Nexus one ini google menunjukkan kemampuan android yang sebenarnya dan hasilnya luar biasa, tanggapan pasar begitu bagus. Hal inilah yang mendorong minat perusahaan-perusahaan *Smartphone* bersemangat untuk membuat *Smartphone* dengan OS Android. Dan hasilnya dapat dilihat saat ini android menguasai pasar smartphone dunia dengan dipimpin oleh Samsung dengan seri galaxynya. Dapat dilihat di Tabel 2.1 merupakan produksi handphone yang ada dipasaran.

Android pertama kali dikembangkan oleh perusahaan bernama Android Inc., dan pada tahun 2005 di akuisisi oleh raksasa Internet Google. Android dibuat dengan basis kernel Linux yang telah dimodifikasi, dan untuk setiap release-nya diberi kode nama berdasarkan nama hidangan makanan. Android adalah sistem operasi Mobile Phone berbasis Linux. Android bersifat open source yang source code-nya diberikan secara gratis bagi para pengembang untuk menciptakan aplikasi mereka agar dapat berjalan di Android.

Tabel 2.1 Produksi Handphone Android 2015

Worldwide Smartphone Operating System 2011 and 2015 Market Share and 2011-2015 Compound Annual Growth Rate			
Operating System	2011 Market Share	2015 Market Share	2011-2015 Unit CAGR
Android	38.9%	43.8%	23.7%
BlackBerry OS	14.2%	13.4%	18.3%
Symbian	20.6%	0.1%	-68.8%
iOS	18.2%	16.9%	17.9%
Windows Phone 7/Windows Mobile	3.8%	20.3%	82.3%
Others	4.3%	5.5%	27.6%
Total	100.0%	100.0%	20.1%

³ Haryanto, Agus 2014. *Android Fast Track: Belajar Membuat Aplikasi Android dengan Mudah dan Cepat*, Agus Haryanto Hal 9.

Antarmuka pengguna Android didasarkan pada manipulasi langsung, menggunakan masukan sentuh yang serupa dengan tindakan di dunia nyata, seperti menggesek, mengetuk, mencubit, dan membalikkan cubitan untuk memanipulasi obyek di layar. Kode dengan sumber terbuka dan lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Keunggulan utama Android adalah gratis dan *open source*, yang membuat *Smartphone* Android dijual lebih murah dibandingkan dengan Blackberry atau iPhone meski fitur (*hardware*) yang ditawarkan Android lebih baik.

Beberapa fitur utama dari Android antara lain *WiFi hotspot*, *Multi-touch*, *Multitasking*, GPS, *accelerometers*, *support java*, mendukung banyak jaringan (GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE & WiMAX) serta juga kemampuan dasar handphone pada umumnya. Pada Gambar 2.5 merupakan Gambar lambing android pada umumnya.



Gambar 2.5 lambang android

2.4 Serial Communication⁴

Komunikasi data merupakan pergerakan data dan informasi yang dikodekan dari satu titik ke titik lain melalui peralatan listrik atau elektromagnetik, kabel serat *optic* atau sinyal gelombang mikro. Dengan adanya teknologi komputer yang semakin canggih saat ini, komunikasi data dapat dilakukan dengan mudah dan cepat. Komunikasi data pada umumnya dapat dilakukan dengan dua cara yaitu secara serial dan

⁴ Suyanto, Muhammad 2005. *Pengantar Teknologi Informasi Untuk Bisnis*, Penerbit Andi.

secara paralel. Komunikasi data serial dilakukan dengan mengirimkan dan menerima data 8 bit secara satu per satu, sedangkan komunikasi data paralel adalah komunikasi data 8 bit secara bersamaan atau sekaligus.

Pada dasarnya ada dua jenis komunikasi data serial, yaitu komunikasi data serial sinkron dimana pengiriman *clock* dilakukan secara bersamaan dengan data serial dan komunikasi data serial asinkron dimana pengiriman *clock* dilakukan secara dua tahap, yaitu saat data dikirimkan dan saat data diterima. Perlu diperhatikan bahwa dalam pengiriman data serial semakin jauh jarak kirim maka kemungkinan *noise* atau gangguan semakin besar. Pada komunikasi data serial pada dasarnya yang dikirim adalah tegangan dan kemudian dibaca dalam data bit.^[5]

2.5 Bluetooth

Bluetooth adalah sebuah alat komunikasi tanpa kabel yang mampu menyediakan layanan transfer data dengan jarak jangkauan yang terbatas. Dalam Tugas akhir ini untuk komunikasi Bluetooth kami menggunakan *Bluetooth HC05* dan resistor 2k Ω dengan Resistor 1 k Ω .^[6] Modul *Bluetooth HC-05* adalah sebuah modul Bluetooth SPP (*Serial Port Protocol*) yang mudah digunakan untuk komunikasi serial *wireless* (nirkabel) yang mengkonversi *port* serial ke *Bluetooth*. Modul ini dapat digunakan sebagai slave maupun master. HC-05 memiliki 2 mode konfigurasi, yaitu *AT mode* dan *Communication mode*. *AT mode* berfungsi untuk melakukan pengaturan konfigurasi dari HC-05. Sedangkan *Communication mode* berfungsi untuk melakukan komunikasi bluetooth dengan piranti lain. Dalam penggunaannya, HC-05 dapat beroperasi tanpa menggunakan driver khusus. Untuk berkomunikasi antar Bluetooth, minimal harus memenuhi dua kondisi berikut :

1. Komunikasi harus antara master dan slave.
2. Password harus benar (saat melakukan pairing).

⁵ Suhata 2005. *VB sebagai Pusat Kendali Peralatan Elektronik*, Elex Media Komputindo. Hal 137

⁶ Jubilee, Enterprise 2010. *Teknik Menghemat Baterai*, Pt Elex Media Komputindo. Hal 62.



Gambar 2.6 Bluetooth HC05

Pada Gambar 2.6 merupakan bentuk fisik modul *Bluetooth* HC05. Jarak sinyal dari HC-05 adalah ± 13 meter, dengan kondisi tanpa halangan. Adapun spesifikasi dari HC-05 adalah :

Hardware : Sensitivitas -80dBm (*Typical*). Daya transmit RF sampai dengan +4dBm., Operasi daya rendah 1,8V – 3,6V I/O. ,Kontrol PIO, Antarmuka UART dengan *baudrate* yang dapat diprogram, Dengan antenna terintegrasi. *Software* : Default baudrate 9600, Data bit : 8, Stop bit = 1, *Parity* : No Parity, Mendukung *baudrate* : 9600, 19200, 38400, 57600, 115200, 230400 dan 460800. ^[7]

2.6 Mikrokontroler^[8]

Mikrokontroler adalah sebuah *system computer* yang dibangun pada sebuah keping (chit) tunggal. Jadi, hanya dengan sebuah keping IC saja dapat dibuat sebuah sistem komputer yang dapat dipergunakan untuk mengontrol alat. Saat ini sebagian besar peralatan elektronika dikontrol dengan mikrokontroler, misalnya mesin fax, mesin fotocopy, mesin cuci otomatis, sampai *handphone*. Mikrokontroler merupakan alat elektronika digital yang hanya tahu 2 kondisi level tegangan, yaitu “logika 0” dan “logika 1”. Secara kasar dapat dikatakan bahwa logika 0 (Low) adalah level tegangan antara 0 volt sampai 0,8 volt. Sedangkan logika 1 (High) adalah level tegangan antara 2 volt sampai 5 volt.

Di dalam mikrokontroler semua informasi diproses secara biner (0 dan 1). Hal ini mengharuskan untuk memberikan intruksi-intruksi dengan bilangan biner. Namun, tentunya sangat susah untuk dapat menggunakan bilangan-bilangan biner tersebut dalam memprogram mikrokontroler.

⁷Pengertian Bluetooth, URL : <http://id.wikipedia.org/wiki/Bluetooth>, 13 Mei 2015

⁸ Malik,Ibnu 2009. *Aneka Proyek Mikrokontroler PLC16F84A* , Pt Elex Media Komputindo Kelompok Gramedia. Hal 1

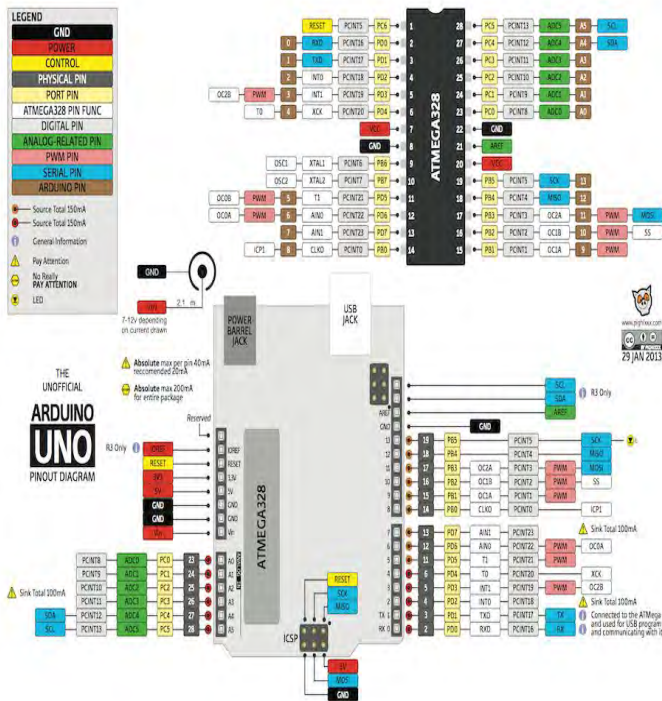
system computer yang dibangun pada Arduino Uno sebenarnya adalah salah satu kit mikrokontroler yang berbasis pada ATmega28. Modul ini sudah dilengkapi dengan berbagai hal yang dibutuhkan untuk mendukung mikrokontroler untuk bekerja. Arduino Uno ini memiliki 14 pin digital input/output, 6 analog input, sebuah resonator keramik 16MHz, koneksi USB, koneksi *power input*, ICSP header, dan sebuah tombol reset.



Gambar 2.7 Mikrokontroller Arduino Uno r3

Secara umum Arduino terdiri dari dua bagian, yaitu: 1. *Hardware* papan *input/output (I/O)* . 2. *Software* Arduino meliputi IDE untuk menulis program, *driver* untuk komunikasi dengan komputer, contoh program dan *library* untuk pengembangan program. Dimana pada perancangan dan pembuatan alat kali ini arduino berfungsi untuk mengatur sejumlah rangkaian dan komponen masukan dan keluaran pada perancangan sistem alat ini.

Fungsi arduino disini akan memberikan program pada lampu lalu lintas yang akan di kontrol saat keadaan jalan raya padat kendaraan bermotor. Sedangkan komponen keluaran yang digunakan adalah komponen *relay* yang nantinya digunakan untuk mengaktifkan *traffic light*. Bentuk fisik arduino uno R3 pada Gambar 2.7 dan pada Gambar 2.8 dijelaskan mengenai skema arduino Uno R3.



Gambar 2.8 Skematik Arduino Uno R3

2.7 Driver Relay

Rangkaian *driver relay* pada perancangan alat ini digunakan untuk mengaktifkan lampu dan sebagai pengontrol lama waktu lampu. *Driver Relay* ini dihubungkan dengan pin I/O arduino, sumber tegangan AC, dan lampu pilot 220VAC. Jadi ketika pin I/O arduino berlogika 1 maka sumber tegangan AC terhubung lampu pilot 220VAC sehingga lampunya nyala. Dan lampu akan berada pada kondisi mati apabila tidak mendapat tegangan masukan dari pin arduino. Komponen rangkaian *driver relay* yang kami gunakan, yaitu *relay* 5V, dioda 1N4007, transistor 2N222, Resistor 1K dan 560 ohm, LED 3mm, pin *screw*.

1. **Relay 5V**^[9]

Relay adalah rangkaian elektronika yang dapat kita gunakan untuk mengendalikan sesuatu dari jarak jauh. *Relay* sendiri merupakan saklar *magnetis* yang paling sering digunakan pada setiap rangkaian elektronika. Dalam dunia elektronika, *relay* sangat berperan penting dalam suatu rangkaian karena dapat mengontrol rangkaian beban arus tinggi dengan arus yang rendah. Dengan menggunakan rangkaian *relay*, tidak perlu untuk mengontrol ataupun mengoperasikan suatu perangkat karena dapat dilakukan dari jarak jauh, sehingga nantinya dapat mempermudah dan memperlancar pekerjaan. Biasanya, rangkaian *relay* terdapat pada televisi, *transmitter*, *sound system*, rangkaian bel, lampu taman otomatis, pengendali peralatan listrik dan perangkat elektronika lainnya. *Relay* yang digunakan pada rangkaian ini adalah *relay* dengan 5 kaki. Dibawah ini pada Gambar 2.9 adalah Gambar dari bentuk fisik *relay*.



Gambar 2.9 *Relay* 5 Kaki

2.8 **Power supply**

Power supply adalah alat yang berfungsi untuk menyalurkan energi listrik atau bentuk energi jenis apapun yang sering digunakan untuk menyalurkan energi listrik. Secara prinsip rangkaian *power supply* yang digunakan pada alat ini bertugas untuk memberikan tegangan input pada Arduino, kipas pendingin dan *driver relay*. Tegangan input yang dibuat yaitu 12V, 9V dan 5V. Tegangan 12V merupakan tegangan yang berfungsi untuk men-*supply* kipas pendingin. Tegangan 9V merupakan tegangan yang masuk pada arduino, karena tegangan input yang disarankan berkisar 7V-12V. jika disuplai oleh tegangan yang lebih

⁹**Rangkaian Relay** , URL: <http://www.rangkaianelektronika.org/rangkaian-relay.htm>, 15 Mei

kecil dari 7V maka *board* arduino menjadi tidak stabil. Sehingga tegangan keluaran dari pin arduino menjadi lebih kecil dari 5V. Jika disuplai lebih dari 12V maka dapat membuat board arduino menjadi panas. Sedangkan tegangan 5V merupakan tegangan masukan pada diver relay. *Power supply* yang dipakai pada alat ini terdiri atas, IC LM 7809, IC LM7805, IC LM7812, TIP 3055, dioda 1N5392 dan kapasitor.

2.9 RTC (*Real Time Clock*)¹⁰

Real Time Clock adalah suatu chip yang mempunyai fungsi untuk penyimpanan waktu dan tanggal. DS1307 merupakan Real Time Clock yang dapat menyimpan data-data waktu seperti detik, menit, tanggal, bulan, hari dalam seminggu, dan tahun valinya hingga 2100. RTC ini mempunyai beberapa spesifikasi yaitu bekerja pada 56-byte, *battery-backed*, RAM *nonvolatile* (NV) RAM untuk penyimpanan. Pengiriman data dari DS1307 ini menggunakan jalur data parallel yang memiliki antarmuka *Serial Two-wire* (I2C), keunggulan dari DS1307 ini juga dapat mendeteksi otomatis kegagalan daya (*power fail*). Konsumsi daya yang digunakan oleh RTC ini adalah kurang dari 500nA jika menggunakan mode baterai cadangan dengan operasional osilator. DS1307 membutuhkan sebuah Kristal 32,768 kHz untuk clock, dan membutuhkan Vcc sebesar 5 volt, dan Vbat sebesar 3volt, dengan dua buah resistor *pull-up* pada *output* SDA(*Serial Data*)&SCL(*Serial Clock*) yang terhubung dengan mikrokontroler. Vbat disini digunakan untuk mensuplay tegangan pada saat tidak ada tegangan dari vcc.

¹⁰RTC DS1307, URL : <http://bekoy.wordpress.com/pemrogaman-rtc-ds1307-menggunakan-cv-avr/> , 19 juni 2015

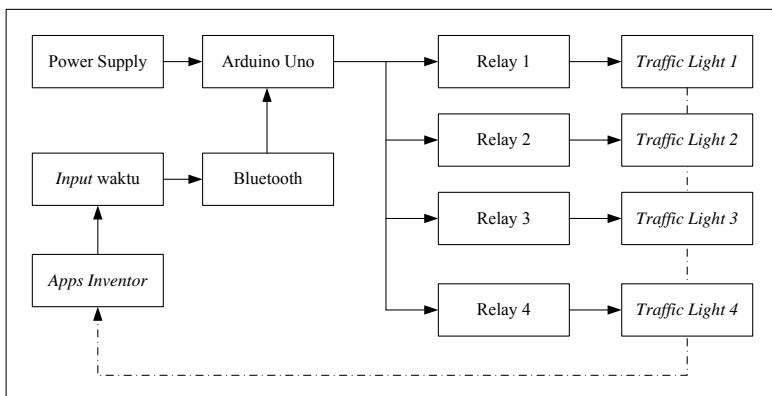
BAB III

PERENCANAAN DAN PEMBUATAN

Pada bab ini berisi bagaimana tahapan yang dilakukan dalam perencanaan dan pembuatan tugas akhir. Penjelasan diawali dengan penjelasan blok fungsional sistem secara keseluruhan, kemudian perancangan perangkat keras, dan diakhiri perancangan perangkat lunak. Adapun perancangan perangkat keras dilakukan pada perancangan mekanik dan elektronik.

Untuk membuat rancang bangun sistem pengaturan lampu lalu lintas yang dikendalikan mikrokontroler dengan menggunakan aplikasi di android secara nirkabel, diperlukan bagian pokok yaitu :

1. Mikrokontroler sebagai unit pengontrol untuk lampu lalu lintas baik *time schedule* dan penambahan waktu.
2. *Driver relay* berfungsi membatasi sinyal digital dengan tegangan bolak-balik untuk mengaktifkan lampu lalu lintas.
3. *Bluetooth* sebagai serial komunikasi antara mikrokontroler dan *handphone* android.
4. *MIT Apps Inventor* sebagai aplikasi antarmuka di *handphone* android untuk mengontrol lampu lalu lintas.



Gambar 3.1 Diagram alat secara keseluruhan

Pada Gambar 3.1 merupakan diagram cara kerja alat secara keseluruhan. Prinsip kerja alat ini adalah *power supply* memberikan sumber tegangan ke arduino uno R3. Setelah mendapatkan sumber

tegangan maka arduino uno R3 akan mengeksekusi program. Lalu menghasilkan keluaran berupa tegangan pada pin. Tegangan keluaran yang dihasilkan dari pin arduino uno R3 akan mengaktifkan *relay* yang digunakan untuk mengontrol penyalan lampu lalu lintas. *Relay* yang aktif akan menyebabkan perubahan posisi saklar sehingga sumber tegangan bolak-balik di *common relay* terhubung dengan lampu lalu lintas di *NO relay* dan lampu lalu lintas menjadi nyala. Waktu nyala lampu lalu lintas dimonitoring melalui aplikasi *apps inventor* di android ketika kondisi jalan padat dengan volum kendaraan maka android mengirimkan masukan data *string* berupa angka ke arduino uno R3 melalui komunikasi serial dengan *Bluetooth HC05*.

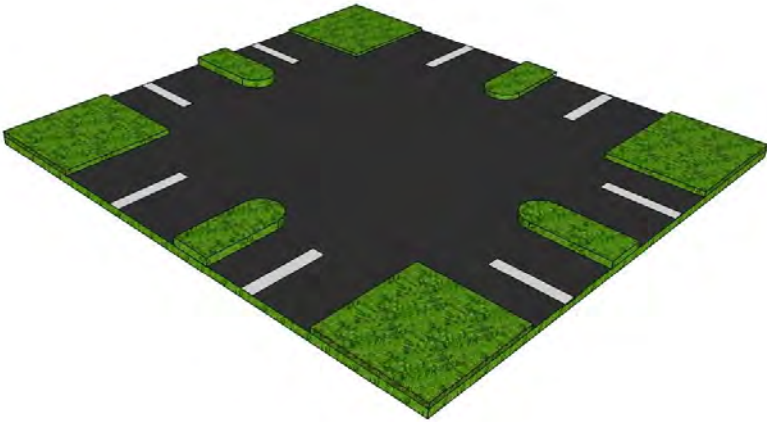
Data yang diterima *Bluetooth HC05* diubah arduino uno R3 menjadi data *integer* dan diolah menjadi masukan *time delay* lampu lalu lintas pada program. Lalu program dieksekusi dan waktu nyala lampu lalu lintas lampu hijau bertambah dan lampu merah lainnya bertambah. Waktu nyala lampu lalu lintas kembali normal setelah kondisi jalan menjadi lancar semua data masukan pada lampu lalu lintas diubah ke *time schedule* awal.

3.1 Perancangan Perangkat Mekanik

Untuk memahami persoalan desain alat untuk mengoperasikan rancang bangun sistem pengaturan lampu lalu lintas yang dikendalikan mikrokontroler dengan aplikasi di android secara nirkabel maka perancangan mekanik yang dibuat pada tugas akhir ini berupa *layout* jalan simpang empat, kotak penyimpanan alat elektronik, dan kotak akrilik lampu dan tiangnya.

3.1.1 Layout jalan simpang empat

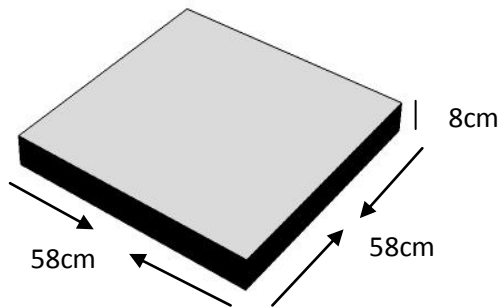
Dibawah ini merupakan rancangan mekanik yang terbuat dari papan kayu yang digunakan sebagai *layout* lampu lalu lintas. pada Gambar 3.2, desain lampu lalu lintas ini diadopsi dari simpang empat jalan semolowaru yang ada di Surabaya. Jalan simpang empat ini merupakan salah satu simpang empat yang sering terjadi kepadatan lalu lintas. Alat dan bahan untuk membuat *layout* Gambar 3.3 dibawah terdapat sebuah papan kayu dengan ukuran 60cm x 60cm x 3cm yang dilapisi dengan kain berudu dan sebuah kertas karton tebal untuk pembuatan jalannya. Lebar satu sisi jalan adalah 14cm.



Gambar 3.2 *layout* jalan simpang empat

3.1.2 Kotak penyimpanan alat elektronik

Pada Gambar 3.3 dibawah ini adalah rancangan mekanik sebuah kotak yang terbuat dari papan kayu dengan ukuran 58cm x 58 cm x 8 cm yang dilapisi cat hitam. Fungsinya sebagai tempat penyimpanan alat elektronik dari peralatan tugas akhir. Diantaranya *microcontroller* arduino uno, *power supply* dan rangkaian *driver relay*.

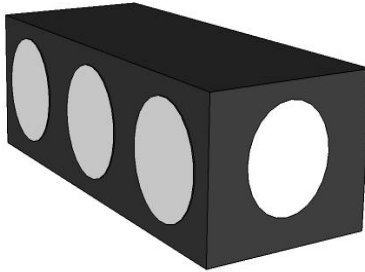


Gambar 3.3 Kotak penyimpanan alat elektronik

3.1.3 Kotak penempatan lampu dan tiang penyangga

Untuk pembuatan lampu lalu lintas pada Gambar 3.4, kami menggunakan sebuah kotak akrilik yang dicat hitam untuk penempatan lampu dengan ukuran 10,5cm x 3,5cm x 4cm serta tiga lubang lingkaran dengan diameter 2,5 cm untuk lampu pilot merah, kuning, hijau.

Pada Gambar 3.5 dibawah ini untuk pembuatan tiang lampu lalu lintas kami menggunakan pipa pvc ukuran 1/8 inchi dengan tinggi 11 cm.



Gambar 3.4 Kotak akrilik lampu lalu lintas



Gambar 3.5 Tiang penyangga lampu lalu lintas

3.2 Perancangan Perangkat Keras (*Hardware*)

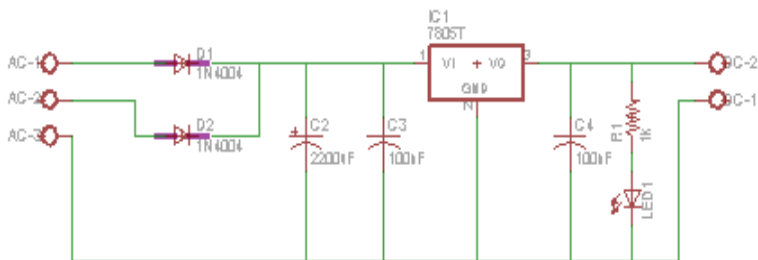
Untuk memahami persoalan sistem kerja elektrik pada alat maka perancangan perangkat keras terdiri dari perancangan rangkaian *power supply* sebagai penyedia sumber tegangan pada alat, perancangan *driver relay* sebagai pengatur *on-off* lampu, dan perancangan komunikasi *Bluetooth* HC05 sebagai penghubung antara *handphone* android dan arduino uno dengan komunikasi serial.

3.2.1 Perancangan *Power supply*

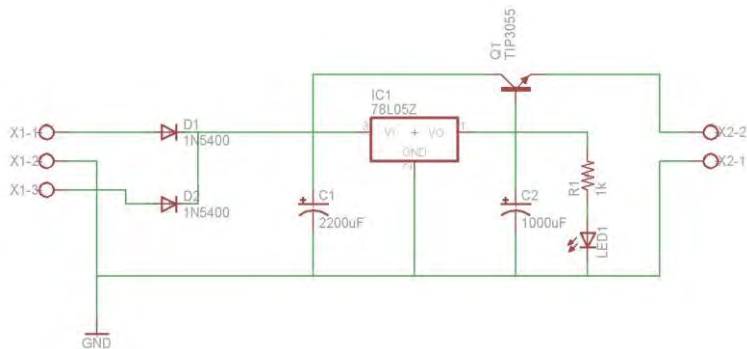
Power supply yang dirancang dalam tugas akhir ini menggunakan 2 *output* tegangan saja dengan 2 terminal. *Power supply* disini berfungsi sebagai penyedia sumber tegangan untuk sistem, yang tersusun dari komponen-komponen, yaitu :

1. Dioda 1N5392 sebagai *rectifier* atau penyearah untuk mengubah tegangan arus bolak-balik (AC) menjadi tegangan arus searah (DC).
2. Kapasitor 2200uF/35V dan 100nF sebagai *filter* untuk meratakan tegangan dari *rectifier*.
3. IC 7805 dan IC 7809 sebagai *regulator* untuk mengubah tegangan menjadi tegangan keluaran yang stabil.

Output tegangan 12 Volt dari Trafo CT 2A diregulasi menggunakan *IC Regulator* 7809 dan 7805 menjadi tegangan 9 Volt dan 5 Volt. Tegangan 9 Volt digunakan untuk mengaktifkan Arduino Uno R3 sedangkan tegangan 5 Volt digunakan untuk mengaktifkan Rangkaian *Driver Relay*. Pada Gambar 3.6 dan 3.7 dijelaskan perancangan skematik *power supply* tegangan 5 Volt dan 9 Volt yang dibuat di *eagle*. Pada Gambar 3.8 dijelaskan bentuk fisik rangkaian *power supply* yang telah dibuat pada *Dote PCB* dan pengujian rangkaian *power supply* tegangan 5 Volt dan 9 Volt pada *ProjectBoard*.



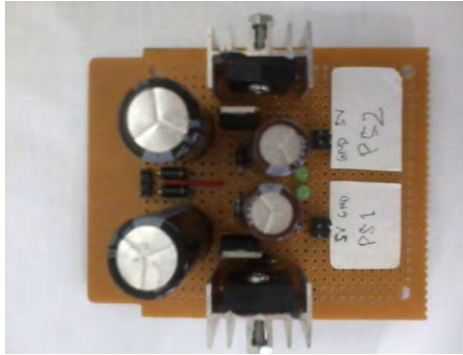
Gambar 3.6 Skematik Rangkaian *Power supply* 9Volt



Gambar 3.7 Skematik Rangkaian *Power supply* 5Volt



(a)

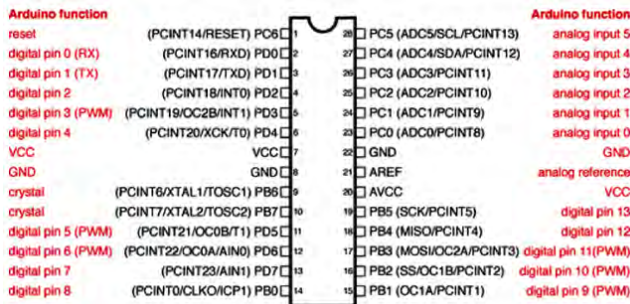


(b)

Gambar 3.8 Bentuk fisik rangkaian (a) *power supply* 9Volt dan (b) *power supply* 5Volt

3.2.2 Perancangan Mikrokontroler

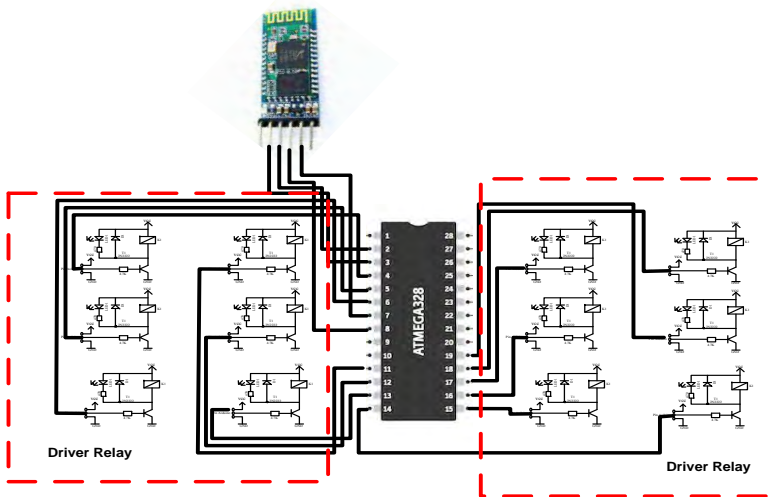
Perangkat utama yang digunakan oleh perancangan alat ini adalah *microcontroller* arduino Uno R3. Arduino Uno adalah sebuah *board* mikrokontroler yang didasarkan pada ATmega 328. Arduino Uno mempunyai 14 pin digital input/output, 6 input analog, sebuah osilator Kristal 16MHz, sebuah koneksi USB, sebuah power jack, sebuah ICSP (*In Circuit Serial Programming*) header dan sebuah tombol reset. Pada Gambar 3.9 merupakan susunan pin ATmega 328.



Gambar 3.9 Konfigurasi Pin ATmega 328/Arduino Uno

Pada Gambar 3.10 dibawah ini adalah Gambar skematik mikrokontroler yang kami gunakan pada rancang bangun pengontrol

lampu lalu lintas. Kami menggunakan beberapa pin ATmega 328 yang dihubungkan dengan *driver relay* yaitu 4, 5, 6, 11, 12, 13, 14, 15, 16, 17, 18, dan 19. Sedangkan *Bluetooth* HC05 dihubungkan dengan pin 2 dan 3 pada ATmega 328.

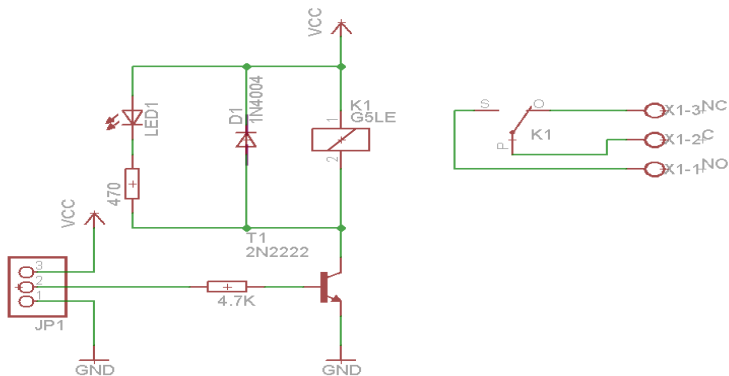


Gambar 3.10 Skematik mikrokontroler

3.2.3 Perancangan *Driver Relay*

Pada tugas akhir ini rangkaian *driver relay* dibutuhkan dalam rangkaian elektronika sebagai eksekutor sekaligus penghubung antara lampu lalu lintas dan sistem kendali arduino uno yang berbeda sistem *power supply* nya. Rangkaian *driver relay* memiliki kondisi saklar atau kontaktor dalam 3 posisi, yaitu NO, NC, CO. Kondisi NO adalah kondisi yang terjadi saat *relay coil* mendapat tegangan sumber. Kondisi NC adalah kondisi yang terjadi saat *relay coil* tidak mendapat tegangan sumber. Kondisi CO adalah kondisi perubahan armatur saklar *relay* dari posisi NC ke NO dan sebaliknya. Pada bagian NO dihubungkan ke lampu lalu lintas, bagian NC tidak dihubungkan, dan bagian C atau *common* dihubungkan ke sumber tegangan bolak-balik. Sehingga rangkaian *driver relay* berfungsi untuk mengontrol nyala lampu lalu

lintas yang penyalurannya diatur dari sistem pengendali arduino uno. Pada Gambar 3.11 adalah skematik dari *driver relay*. Dan pada Gambar 3.12 dijelaskan bentuk fisik dari *driver relay*.



Gambar 3.11 Skematik *relay driver*



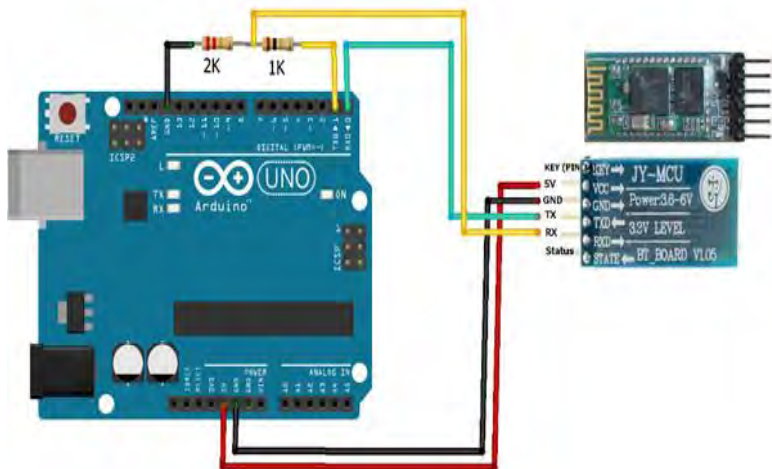
Gambar 3.12 Bentuk fisik *relay driver*

3.2.4 Komunikasi Serial *Bluetooth HC-05*

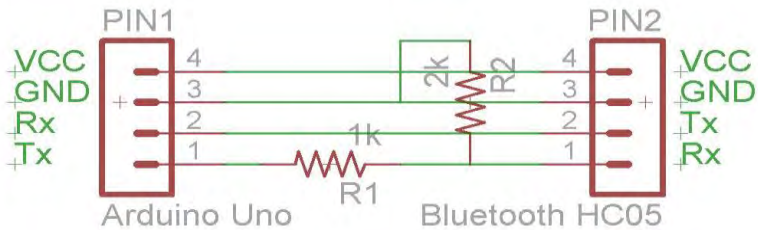
Untuk komunikasi serial dalam tugas akhir ini menggunakan *Bluetooth HC-05* yang sudah ada *Bluetooth serial module transceiver*. *Bluetooth HC-05* dapat dikoneksikan sebagai *slave*, ataupun sebagai *master*. Dalam penggunaannya, *Bluetooth HC-05* ini diatur terlebih dahulu sebagai *slave* dengan menggunakan *AT Commands*. Pengaturan *slave mode* pada *AT Commands* dengan mengetik “AT+ROLE = 0” di

serial monitor arduino uno maka akan tampil “OK” sebagai respon *Bluetooth* HC-05 sehingga *Bluetooth* HC-05 sudah menjadi *slave mode*.

Dengan begitu proses pengaturan mulai pengiriman dan penerimaan data berupa angka untuk lampu lalu lintas dari aplikasi *app inventor 2* di *handphone* android dapat dilakukan secara nirkabel. Dalam pemasangannya pin RX arduino uno dengan pin TX *Bluetooth* HC-05 dan pin TX *Bluetooth* HC-05 dengan pin RX Arduino uno. Pada pin TX arduino menghasilkan tegangan keluaran sebesar 5V sehingga dalam pemasangan ke pin RX *Bluetooth* HC-05 memerlukan *voltage divider* supaya menjadi 3.3V. Pada Gambar 3.13 dijelaskan Perancangan pemasangan antara *Bluetooth* HC-05 modul dengan Arduino Uno R3. Pada Gambar 3.14 dijelaskan perancangan skematik rangkaian komunikasi serial *Bluetooth* HC-05 modul dengan Arduino Uno R3.



Gambar 3.13 Komunikasi serial *Bluetooth* HC-05 dan Arduino Uno R3



Gambar 3.14 Skematik rangkaian serial *Bluetooth* HC-05 dan Arduino Uno R3

3.2.5 Perancangan RTC DS1307

Pada perancangan RTCDS1307 ini bertujuan untuk menyamakan nyala waktu lampu hijau sesuai dengan *time schedule* pada persimpangan lalu lintas di jalan semolowaru. Adapun waktu lampu hijau yang digunakan pada *time schedule* dapat dilihat pada Tabel 3.1 dibawah ini.

Tabel 3.1 *Time Schedule* Simpang Empat Jl Semolowaru

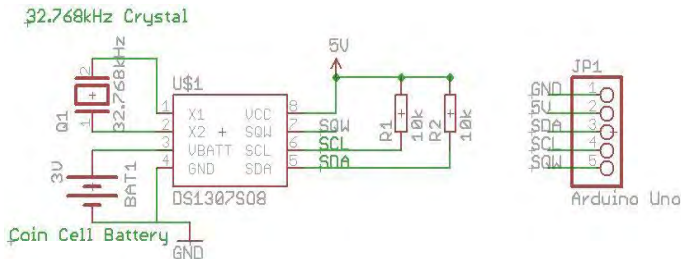
Plan	Utara(detik)	Timur(detik)	Barat(detik)	Selatan(detik)
0	0	0	0	0
1	19	40	25	33
2	25	40	25	33
3	18	26	25	33
4	18	38	35	36
5	18	30	25	40
6	15	20	15	10
7	15	38	25	30

Pada Tabel 3.1 merupakan Tabel mengenai lamanya lampu hijau disetiap *plan*, sedangkan untuk setiap jam pada *plan* dapat dilihat dalam Tabel 3.2.

Tabel 3.2 Jadwal Jam Pada *Plan*

Plan	Jadwal jam
0	-persiapan-
1	09.00 – 11.00
2	11.00 – 15.00
3	15.00 – 19.00
4	19.00 – 21.00
5	21.00 – 22.00
6	22.00 – 05.00
7	05.00 – 09.00

Pada plan 0 ialah plan dimana saat lampu lalu lintas persiapan untuk dinyalakan. Pada plan 0 lampu kuning disetiap simpang akan berkedip secara bergantian selama 120 detik. Dalam RTC DS1307 ini mempunyai 8 pin. Perancangan pin 4,5,6,dan 8 akan masuk ke board arduino menuju input analog yaitu A3,A4,A5,dan A2. Sedangkan untuk pin 3 akan mendapatkan *supply* dari baterai 3 volt. Pada pin 1,dan 2 menuju ke Kristal 32,768 kHz.



Gambar 3.15 Skematik RTC



Gambar 3.16 Gambar fisik RTC

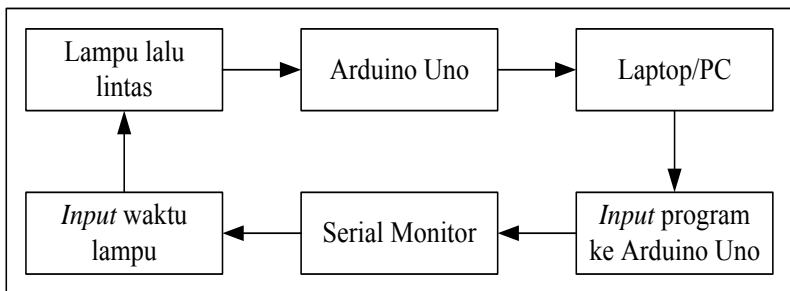
Pada Gambar 3.15 merupakan gambar skematik dari RTC DS1307. Dari skematik tersebut dapat dilihat setiap pin RTC DS 1307 terhubung dengan komponen yang berbeda-beda. Pada Gambar 3.16 adalah bentuk fisik dari RTC DS1307 yang sudah kami buat.

3.3 Perancangan Perangkat Lunak (*Software*)

Perancangan perangkat lunak (*software*) pada rancang bangun pengaturan lampu lalu lintas berbasis mikrokontroler yang dikendalikan dengan aplikasi di android secara nirkabel ini meliputi pembuatan program dengan menggunakan aplikasi IDE Arduino 1.0.5 dan aplikasi *online Google App inventor*.

3.3.1 Perancangan program *traffic light*

Perancangan program pengaturan *time schedule* lampu lalu lintas dengan tujuan untuk mengurangi kepadatan kendaraan bermotor di jalan simpang empat. Dimana pengaturan ini dilakukan secara nirkabel dengan menambahkan waktu lampu hijau. Diagram blok pembuatan program dapat dilihat pada Gambar 3.17 dibawah ini.



Gambar 3.17 Diagram Blok Penambahan *Time Delay* Lampu Lalu Lintas

Pada diagram blok diatas lampu lalu lintas disambungkan dengan arduino uno R3. Selanjutnya arduino akan disambungkan dengan *PC/Laptop* untuk menjalankan program penambahan *time delay* lampu lalu lintas. Selanjutnya pada tampilan IDE Arduino Uno 1.0.5 klik *serial monitor* dipojok kanan atas. Serial monitor dapat dilihat pada Gambar 3.18 dibawah ini :



Gambar 3.18 Serial Monitor Pada Tampilan Program

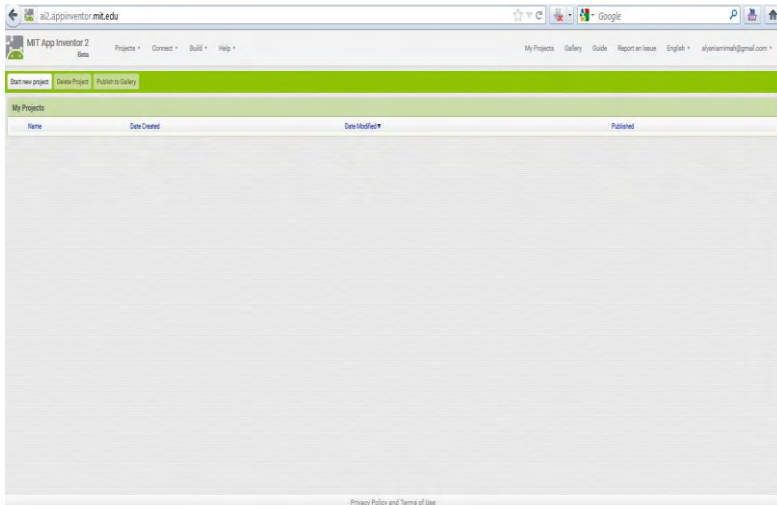
Pada *point* dibawah ini adalah alogaritma dari program pengaturan nyala lampu lalu lintas. Proses penambahan waktu *delay* pada alogaritma di bawah bertambah saat kondisi padat kendaraan di salah satu persimpangan jalan. Jika setelah penambahan waktu *delay* kondisi jalan masih padat maka proses akan berulang kembali ke proses awal, alogaritmanya seperti dibawah ini :

1. Plant 0, Lampu kuning menyala blink selama 120 detik sebagai tahap persiapan.
2. Jika tombol tekan start di android, maka blink lampu kuning berpindah time schedule lampu lalu lintas.
3. Jika jam antara pukul 09.00 sampai 11.00, maka time schedule yang beroperasi adalah plant 1 bekerja pada lampu lalu lintas sebagai time schedule.
4. Jika jam antara pukul 11.00 sampai 15.00, maka time schedule yang beroperasi adalah plant 2 bekerja pada lampu lalu lintas sebagai time schedule.
5. Jika jam antara pukul 15.00 sampai 19.00, maka time schedule yang beroperasi adalah plant 3 bekerja pada lampu lalu lintas sebagai time schedule.

6. Jika jam antara pukul 19.00 sampai 21.00, maka time schedule yang beroperasi adalah plant 4 bekerja pada lampu lalu lintas sebagai time schedule.
7. Jika jam antara pukul 21.00 sampai 22.00, maka time schedule yang beroperasi adalah plant 5 bekerja pada lampu lalu lintas sebagai time schedule.
8. Jika jam antara pukul 22.00 sampai 05.00, maka time schedule yang beroperasi adalah plant 6 bekerja pada lampu lalu lintas sebagai time schedule.
9. Jika jam antara pukul 05.00 sampai 09.00, maka time schedule yang beroperasi adalah plant 7 bekerja pada lampu lalu lintas sebagai time schedule.
10. Jika ada penambahan waktu delay pada plant 1, 2, 3, 4, 5, 6, atau 7 maka input waktu delay ke lampu hijau 1, 2, 3, atau 4 pada persimpangan empat jalan.
11. Setelah selesai penambahan waktu delay, tekan tombol reset sehingga waktu delay lampu hijau kembali ke settingan plant awal.

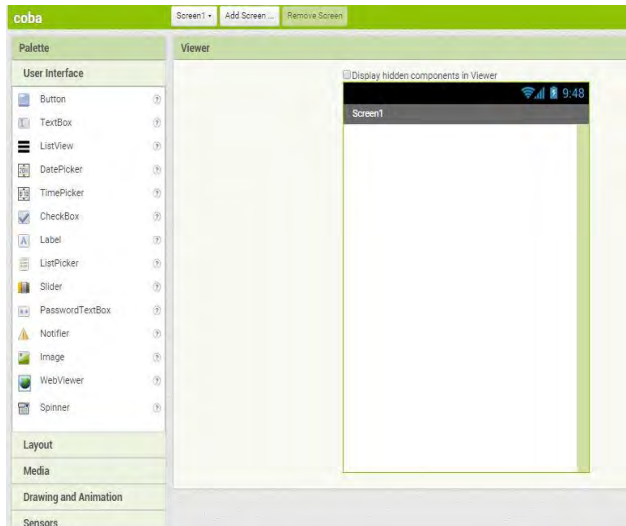
3.3.2 Perancangan Perangkat Lunak Menggunakan Google App Inventor

Perancangan *interface* antara *handphone* android dan lampu lalu lintas yang diatur oleh mikrokontroler arduino uno R3 bertujuan untuk menambah waktu tunda lampu lalu lintas saat keadaan di simpang empat terjadi kepadatan kendaraan. Penambahan waktu tunda menggunakan program yang telah dibuat di *google app inventor* yang terhubung *wireless* melalui komunikasi serial *bluetooth*. Aplikasi dirancang agar tampilan yang dihasilkan terlihat menarik, mudah dipahami dan dioperasikan. Perancangan pembuatan aplikasi ini dilakukan secara *online* pada alamat web <http://ai2.appinventor.mit.edu/> dan harus login terlebih dahulu menggunakan akun Gmail. Disini kami menggunakan *app inventor* dengan versi ke 2. Pada Gambar 3.19 tampak tampilan awal *App inventor* setelah *log in*.



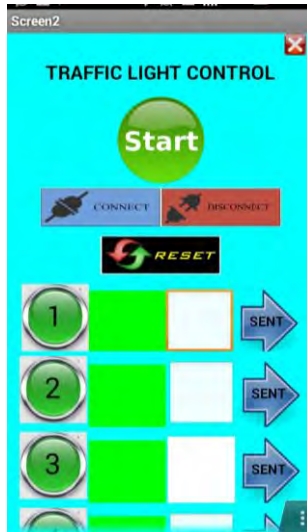
Gambar 3.19 Tampilan awal *MIT App Inventor 2*

Untuk membuat sebuah aplikasi yang baru di *app inventor* diawali dengan menekan tombol *start new project* pada bagian atas di pojok kiri. Setelah menekan tombol *start new project* akan ada pemberitahuan untuk pemberian nama program. Setelah selesai pemberian nama, masuk pada pembuatan tampilan di kotak dialog *Design* untuk menambahkan waktu *delay* pada lampu lalu lintas. Pada Gambar 3.20 merupakan tampilan awal dari *app designer*.



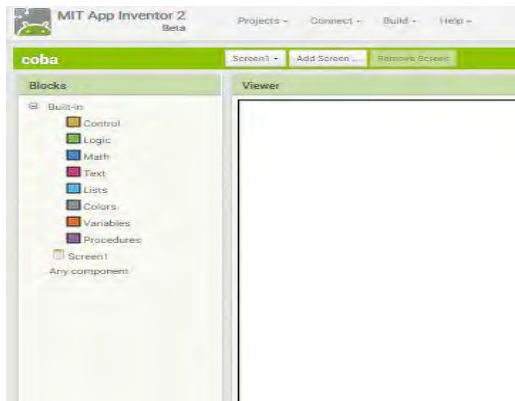
Gambar 3.20 Tampilan *app designer*

Pada Gambar 3.20 tampilan pada screen masih kosong, nantinya pada *screen* tersebut akan di *desain* untuk menyusun komponen dasar di *user interface* seperti *button*, *label*, *textbox*, *image* dan komponen-komponen lainnya. Untuk menambahkan komponen, *user* cukup *drag and drop* pada *screen* yang masih kosong. Pada Gambar 3.21 merupakan hasil *desain* dari rancangan kami untuk membuat aplikasi di *google app inventor*.

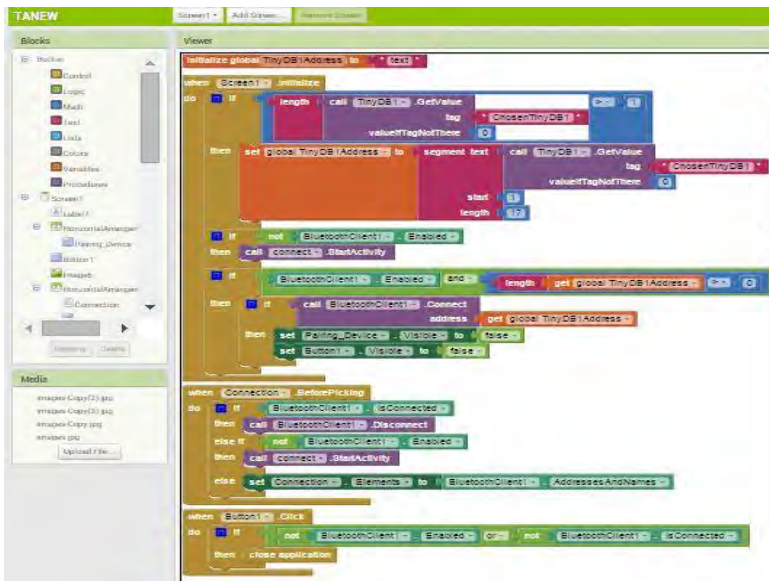


Gambar 3.21 Layout aplikasi

Setelah membuat layout aplikasi yang diinginkan barulah memasukkan program yang berupa blok-blok kode dengan mengeklik *block* pada sisi pojok kanan. Tampilan blok awal sebelum di edit ada pada Gambar 3.22 , dan tampilan blok yang sudah kami buat terlihat pada Gambar 3.23.

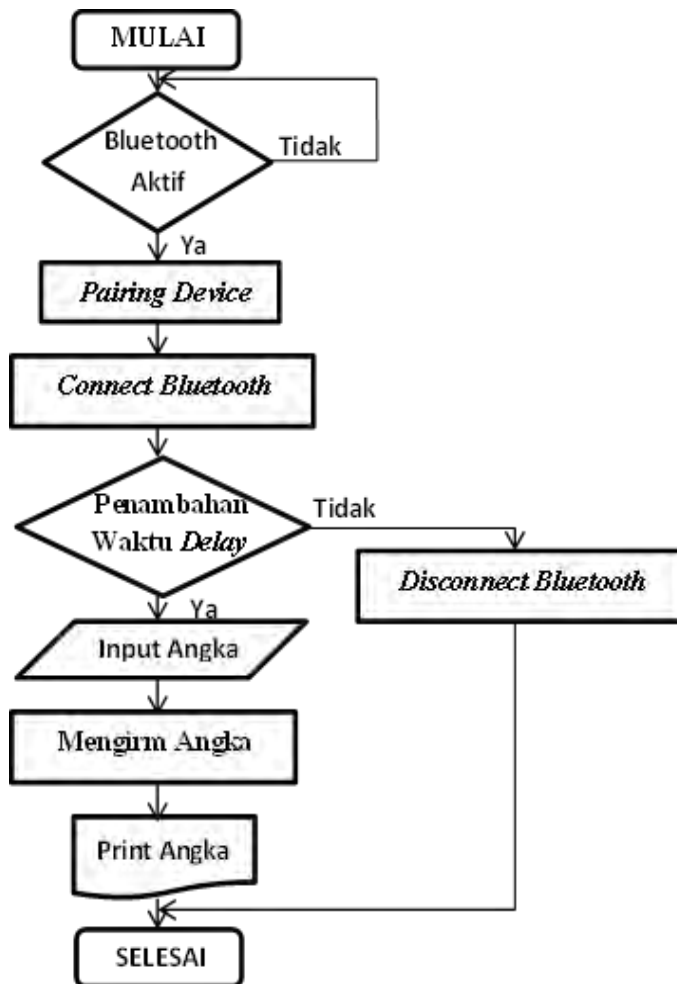


Gambar 3.22 tampilan desain blok awal



Gambar 3.23 *logic blok pengaturan delay time traffic light*

Pada Gambar 3.24 dibawah ini adalah *flowchart* dari sistem kerja dari program pengontrol lampu lalu lintas yang menggunakan *google app inventor*



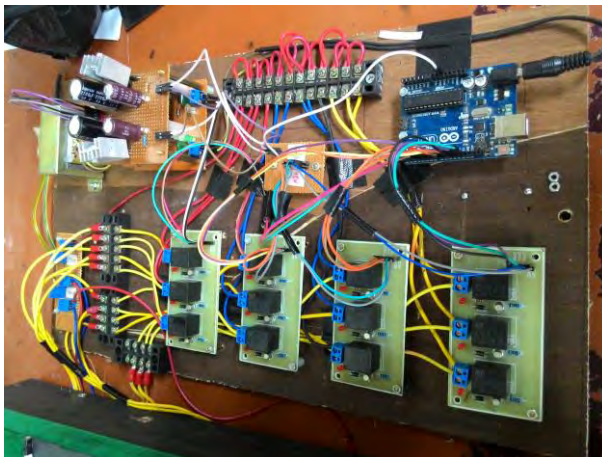
Gambar 3.24 Flowchart Program MIT APP Inventor

BAB IV

PENGUJIAN DAN PENGUKURAN

Untuk mengetahui apakah tujuan-tujuan dari pembuatan alat ini telah tercapai atau belum, maka perlu dilakukannya sebuah pengujian dan analisa terhadap alat yang telah dibuat. Dan sebagai acuan yang tidak terpisahkan adalah adanya proses *evaluasi* sehingga akan dapat dilakukan langkah-langkah positif guna membawa alat ini kearah yang lebih baik. Hasil pengujian tersebut akan dianalisa untuk mengetahui penyebab terjadinya kekurangan dalam sistem. Pengujian tersebut meliputi :

1. Pengujian Perangkat Keras (*Hardware*)
 - a. *Microcontroller* Arduino Uno R3
 - b. Rangkaian *Power supply*
 - c. Rangkaian *Driver Relay*
 - d. *Bluetooth* HC05
 2. Pengujian Perangkat Lunak (*Software*)
 - a. Pengujian *Serial Monitor* Lampu Lalu Lintas
 - b. Pengujian Program *Apps Inventor* dengan *Bluetooth*
 3. Pengujian Alat secara Keseluruhan
- . Alat yang telah selesai dapat dilihat pada Gambar 4.1.



Gambar 4.1 Implementas Rangkaian Kontrol Pengatur *Traffic light*

4.1 Pengujian Perangkat Keras

Perangkat keras (*hardware*) yang dipakai pada tugas akhir kali ini meliputi rangkaian *power supply* dan rangkaian *driver relay*. Berikut adalah pembahasan tentang pengujian *microcontroller* arduino uno R3, rangkaian *power supply*. Rangkaian *Bluetooth* dan rangkaian *driver relay* yang dipakai pada lampu lalu lintas.

4.1.1 Pengujian *Input/Output* Mikrokontroler Arduino Uno

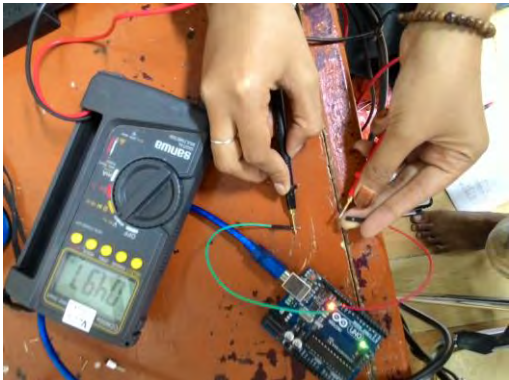
Pengujian ini dilakukan terhadap *board* Arduino Uno. Pengujian dilakukan pada 14 pin digital Arduino Uno dengan cara mengeset outputnya pada logika 1 dan 0. Tujuan dari pengujian Input/Output pada arduino uno yang kami gunakan ini adalah untuk mengetahui berapakah tegangan *output* pada setiap pin arduino jika diberi *input low* dan *high*. Setelah kami uji data yang didapat adalah seperti pada Tabel 4.1.

Tabel 4.1 Pengujian I/O Arduino Uno

Nomer Pin	Logic	Tegangan Terukur (V)	Logic	Tegangan Terukur (V)
0	1	4.98	0	0.00
1	1	4.98	0	0.00
2	1	4.98	0	0.00
3	1	4.98	0	0.00
4	1	4.98	0	0.00
5	1	4.98	0	0.00
6	1	4.98	0	0.00
7	1	4.98	0	0.00
8	1	4.98	0	0.00
9	1	4.98	0	0.00
10	1	4.98	0	0.00
11	1	4.98	0	0.00
12	1	4.98	0	0.00
13	1	4.98	0	0.0

Setelah pengambilan data input/output pada pin arduino selesai, dari data yang kami peroleh dapat disimpulkan bahwa arduino ini dapat bekerja dalam keadaan high dengan tegangan 4,98 volt. Angka tersebut

mendekati 5 volt sesuai yang ada didatasheet arduino uno. Pada Gambar 4.2 merupakan pengujian input output mikrokontroler arduino uno.



Gambar 4.2 Pengujian *Input* dan *Output* Mikrokontroler Arduino Uno

4.1.2 Pengujian Rangkaian *Power supply*

Pada perancangan tugas akhir kali ini rangkaian *power supply* digunakan sebagai sumber tegangan yang digunakan untuk mensuplai ke *microcontroller* arduino uno R3 dan rangkaian *driver relay*. Rangkaian *power supply* bertugas untuk memberi tegangan masukan sebesar 9V dan 5V. Jadi, pada kesimpulannya rangkaian *power supply* berikut mempunyai tegangan masukan sebesar 9V dan 5V. Yang akan diuji pada rangkaian *power supply* adalah besar tegangan masukan pada *power supply* dan tegangan keluaran pada *power supply*. Serta dihitung persentase kesalahan pada rangkaian *power supply*.

Berikut pada Tabel 4.2 adalah Tabel pengujian rangkaian *power supply* 5V pada perancangan alat :

Tabel 4.2 Pengujian Rangkaian *Power supply* 5V

Pengukuran Ke -	Tegangan Masukan	Tegangan Keluaran			
	Vin (Volt)	Vout 1 Tanpa Beban	Vout 2 Tanpa Beban	Vout 1 Dengan Beban	Vout 2 Dengan Beban
1	13,66V	4.90 V	4,93V	4.34 V	4.32 V
2	13,65V	4.95 V	4,94V	4.35 V	4.33 V

Pengukuran Ke -	Tegangan Masukan	Tegangan Keluaran			
	Vin (Volt)	Vout 1 Tanpa Beban	Vout 2 Tanpa Beban	Vout 1 Dengan Beban	Vout 2 Dengan Beban
3	13,66V	4.90 V	4,92V	4.35 V	4.33 V
4	13,65V	4.91 V	4,91V	4.36 V	4.34 V
5	13,65V	4.92 V	4,96V	4.36 V	4.35 V
6	13,65V	4.97 V	4,92V	4.37 V	4.35 V
7	13,66V	4.98 V	4,91V	4.37 V	4.35 V
8	13,66V	4.96V	4,92V	4.38 V	4.35 V
9	13,66V	4.96 V	4,92V	4.38 V	4.36 V
10	13,66V	4.98 V	4,92V	4.38 V	4.36 V
V rata-rata	13,66 V	4.943 V	4,925V	4.364V	4.343 V

Pengujian tanpa beban dilakukan dengan mengukur keluaran dengan *Volt* meter, kemudian diukur persentase error.

% Kesalahan untuk *Power supply* +5V pertama tanpa beban :

$$= 5 - 4,943 / 5 \times 100 \%$$

$$= 1,14 \%$$

% Kesalahan untuk *Power supply* +5V kedua tanpa beban :

$$= 5 - 4,925 / 5 \times 100 \%$$

$$= 1,5 \%$$

% Kesalahan untuk *Power supply* +5V pertama dengan beban :

$$= 5 - 4,364 / 5 \times 100 \%$$

$$= 12,72 \%$$

% Kesalahan untuk *Power supply* +5V kedua dengan beban :

$$= 5 - 4,343 / 5 \times 100 \%$$

$$= 13,4\%$$

Berikut pada Tabel 4.3 adalah Tabel pengujian rangkaian *power supply* 9V pada perancangan alat :

Tabel 4.3 Pengujian Rangkaian *Power supply* 9V

Pengukuran Ke -	Tegangan Masukan	Tegangan Keluaran	
	<i>V_{in}</i> (Volt)	<i>V_{out}</i> Tanpa Beban	<i>V_{out}</i> Dengan Beban
1	13.4V	9V	9V
2	13.4V	9V	9V
3	13.4V	9V	9V
4	13.4V	9V	9V
5	13.4V	9V	9V
6	13.4V	9V	9V
7	13.4V	9V	9V
V rata-rata	13.4V	9V	9V

Pengujian tanpa beban dilakukan dengan mengukur keluaran dengan *Volt* meter, kemudian diukur persentase error.

$$\begin{aligned}
 \% \text{ Kesalahan untuk } Power \text{ supply } +9V \text{ tanpa beban :} \\
 &= \frac{9 - 9}{9} \times 100 \% \\
 &= 0\%
 \end{aligned}$$

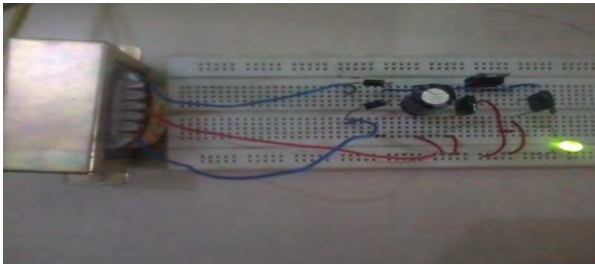
$$\begin{aligned}
 \% \text{ Kesalahan untuk } Power \text{ supply } +9V \text{ dengan beban :} \\
 &= \frac{9 - 9}{9} \times 100 \% \\
 &= 0 \%
 \end{aligned}$$

Setelah kami ambil hasil pengujian *power supply* 5volt didapatkan tegangan rata-rata output yang keluar tanpa beban sebesar 4,99 volt dan yang menggunakan beban 4,3 volt. Dari tegangan rata-rata tersebut dapat diambil persentase error dengan rumus:

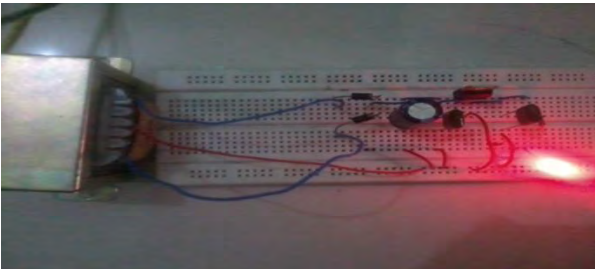
$$\%error = \frac{V_{out} - V_{out \text{ rata-rata}}}{V_{out}} \times 100\%$$

Dari rumus tersebut diperoleh persentase error dari *power supply* pertama 5V tanpa beban sebesar 1,14% dan *power supply* kedua 5V tanpa beban sebesar 1,5% sedangkan dengan *power supply* pertama 5V dengan beban sebesar 12,72 % dan *power supply* 5V kedua sebesar 13,4% . Untuk pengujian *power supply* 9V didapatkan tegangan

rata-rata output yang keluar tanpa beban sebesar 9V dan yang menggunakan beban 9V. Dengan presentase error dari *power supply* 9V tanpa beban sebesar 0% dan dengan beban sebesar 0%. Pada Gambar 4.3 adalah foto saat pengujian *power supply* 5volt pada project board, dan pada Gambar 4.4 adalah foto saat pengujian *power supply* 9 volt di *project board*.



Gambar 4.3 Pengujian rangkaian *power supply* 5V di *Projectboard*



Gambar 4.4 Pengujian Rangkaian *Power supply* 9V di *Projectboard*

4.1.3 Pengujian Rangkaian *Driver Relay*

Rangkaian yang dipakai pada perancangan tugas akhir ini digunakan sebagai *output* yang bertugas untuk mengaktifkan lampu lalu lintas, *Relay* yang dipakai yaitu *relay* dengan 5 kaki. Pengujian pada *relay* dilakukan pada kondisi *relay* dan tegangan pada Vcc. Pengujian diambil dengan 3 lampu lalu lintas dengan 3 *relay* dan *delay* 10 detik. Pengujian ini bertujuan untuk mengetahui bagaimana performa kerja *relay* dalam mengaktifkan lampu. Cara kerja pengujian ialah dengan

menyambungkan setiap *relay* ke 12 input digital yang ada di mikrokontroler arduino uno.

Berikut pada Tabel 4.4 hasil dari pengujian rangkaian *relay* pada perancangan alat pengontrol lampu lalu lintas :

Tabel 4.4 Pengujian Rangkaian *Relay*

Waktu kerja <i>relay</i> (menit)	Kondisi <i>Relay</i>	Vcc
5	Baik	4,3 V
10	Baik	4,3 V
15	Baik	4,3 V
20	Baik	4,3 V
25	Baik	4,3 V
30	Baik	4,3 V
35	Baik	4,3 V
40	Baik	4,3 V
45	Baik	4,3 V
50	Baik	4,3 V

Dari data diatas 12 relay yang digunakan yang digunakan pada perancangan dan pembuatan pengaturan dan monitoring lampu lalu lintas berbasis mikrokontroler yang dikendalikan dengan android secara nirkabel ini dapat disimpulkan *relay* bekerja dengan baik karena setelah 50 menit pengujian *relay* masih bekerja secara normal sesuai nyala lampu dan kondisi baik.

4.1.4 Pengujian Module Bluetooth

Dalam pengujian *module Bluetooth* ini, kita akan mencoba menguji jarak komunikasi antara handphone yang sudah terintegrasi dengan android dan *module Bluetooth* yang sudah tersambung dengan arduino. Tujuannya ialah untuk mengetahui seberapa jauh *Bluetooth* ini bias mengirim data dan menerima data dengan baik. Kami akan melakukan beberapa pengujian, hasilnya akan diketahui pada Tabel 4.5 berikut ini, kami mengambil data berapa jarak maksimal *Bluetooth* ini bisa mengirim data dengan kondisi tanpa halangan :

Tabel 4.5 Pengujian jarak koneksi *Bluetooth*

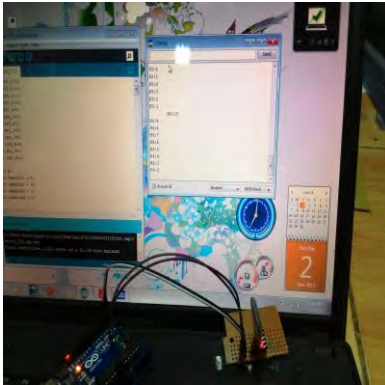
	Pengujian ke				
Jarak (meter)	1	2	3	4	5
2	connect	connect	connect	connect	connect
4	connect	connect	connect	connect	connect
6	connect	connect	connect	connect	connect
8	connect	connect	connect	connect	connect
10	connect	connect	connect	connect	connect
12	connect	connect	connect	connect	connect
14	disconnect	disconnect	disconnect	disconnect	disconnect

Dari data pengujian pada Tabel 4.5 dapat disimpulkan bahwa jarak maksimal *Bluetooth* yang kami gunakan dapat mengirim data dan menerima data adalah antara 0 – 12 meter, setelah jarak tersebut *Bluetooth* tidak dapat bekerja maksimal. Pada Tabel 4.6 merupakan lamanya waktu pengiriman data dari bluetotoh yang kami gunakan.

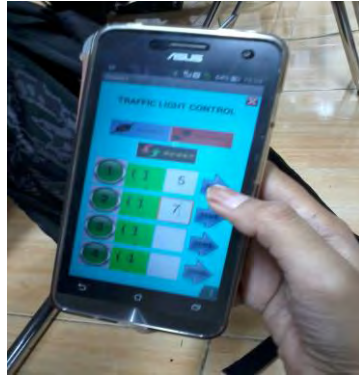
Tabel 4.6 Waktu Pengiriman data *Bluetooth*

Jarak (meter)	Waktu pengiriman data (s)					RATA - RATA
	1	2	3	4	5	
2	1.26	0.96	1.33	1.1	1.49	1.228
4	1.59	1.19	1.39	1.24	1.52	1.386
6	1.68	1.26	1.63	1.47	1.67	1.542
8	1.73	1.3	1.7	1.67	1.81	1.642
10	1.82	1.68	1.84	1.74	2.02	1.82
12	1.89	2.13	2.04	2.25	2.03	2.068
14	0	0	0	0	0	0

Pada Gambar 4.5 adalah foto saat pengujian pengambilan data *Bluetooth*.



(a)



(b)

Gambar 4.5 (a)koneksi *Bluetooth* dengan arduino melalui pc
(b)monitoring data *Bluetooth*.

4.1.5 Pengujian RTC (*Real Time Clock*)

Dalam pengujian RTC (*Real Time Clock*) ini , kami menggunakan LDC sebagai tampilan waktu dari RTC, power supply sebagai sumber tegangan untuk masukan ke arduino, dan RTC DS1307. Tujuan dari pengujian ini adalah untuk mengetahui apakah ada pergeseran waktu antara antara real time pada hand phone dengan waktu dari RTC DS1307. Hasil pengujian yang sudah kami lakukan dapat dilihat pada Tabel 4.7 dibawah ini.

Tabel 4.7 Pengujian RTC

Pengujian ke	Hari 1		Hari 2	
	Jam digital Laptop	RTC	Jam digital Laptop	RTC
1	12:25	12:25	12:25	12:25
2	12:45	12:45	12:45	12:45
3	13:05	13:05	13:05	13:05
4	13:25	13:25	13:25	13:25
5	13:40	13:40	13:40	13:40
6	13:50	13:50	13:50	13:50
7	14:00	14:00	14:00	14:00
8	14:10	14:10	14:10	14:10

Pengujian ke	Hari 1		Hari 2	
	Jam digital Laptop	RTC	Jam digital Laptop	RTC
9	14:20	14:20	14:20	14:20
11	14:40	14:40	14:40	14:40
12	14:50	14:50	14:50	14:50
13	15:00	15:00	15:00	15:00

Dari Tabel 4.7 diatas dapat disimpulkan tidak perbedaan antara jam digital pada lapto dan waktu RTC yang ditampilkan pada LCD. Oleh karena itu dapat diambil kesimpulan bahwa RTC dapat digunakan sebagai acuan karena tidak ada selisih waktu. Pada Gambar 4.6 merupakan foto bagaimana pengujian RTC.



Gambar 4.6 Pengujian RTC

4.2 Pengujian Perangkat Lunak (*Software*)

Perangkat lunak (*software*) yang digunakan pada rancang bangun pengaturan lampu lalu lintas berbasis mikrokontroler yang dikendalikan dengan aplikasi di android secara nirkabel adalah program IDE Arduino 1.0.5. dan *Google App Inventor*

4.2.1 Pengujian *Serial Monitor* Untuk Menambah Waktu Lampu Lalu Lintas

Proses kerja penambahan waktu *delay* lampu hijau, akan bekerja setelah satu looping program itu berjalan. Pertama saat dirasa ada kepadatan kendaraan di ruas jalan pertama maka, polisi akan menambah waktu pada saat program sedang berjalan di ruas jalan pertama. Jika ruas

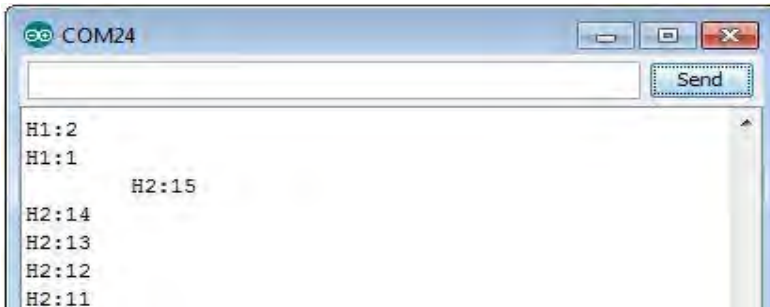
jalan ke dua yang padat dengan kendaraan maka polisi akan menambahkan waktu *delay* di ruas jalan yang kedua. Setelah program berjalan hingga ruas jalan ke empat, maka *looping* yang selanjutnya pada lampu hijau di ruas jalan kedua akan bertambah dan begitu seterusnya. Jika pada lampu lalu lintas tersebut kepadatan kendaraan sudah mulai normal maka lama waktu lampu dapat diatur kembali ke *time schedule* awal, ini adalah Gambar proses penambahan waktu *delay* melalui *serial monitor*. Pada Gambar 4.7 merupakan *serial monitor* saat proses program awal tanpa pemberian input. Pada Gambar 4.8 merupakan *serial monitor* saat proses penambahan waktu pada lampu hijau pertama. Pada Gambar 4.9 merupakan *serial monitor* saat proses penambahan waktu pada lampu hijau kedua. Pada Gambar 4.10 merupakan *serial monitor* saat proses penambahan waktu pada lampu hijau ketiga. Pada Gambar 4.11 merupakan *serial monitor* saat proses penambahan waktu pada lampu hijau keempat. Pada Gambar 4.12 merupakan *serial monitor* saat proses pemberian input *reset*.



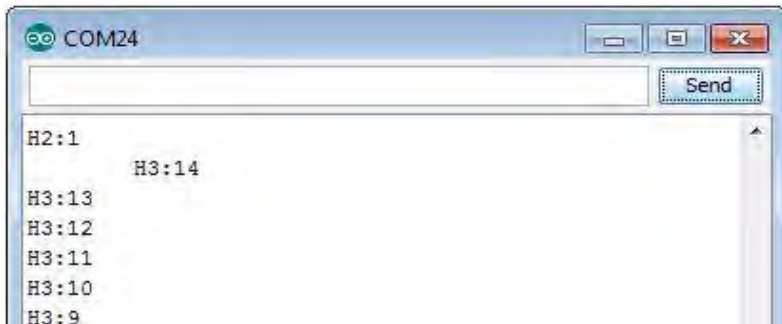
Gambar 4.7 *Serial Monitor* menjalankan *Traffic light*



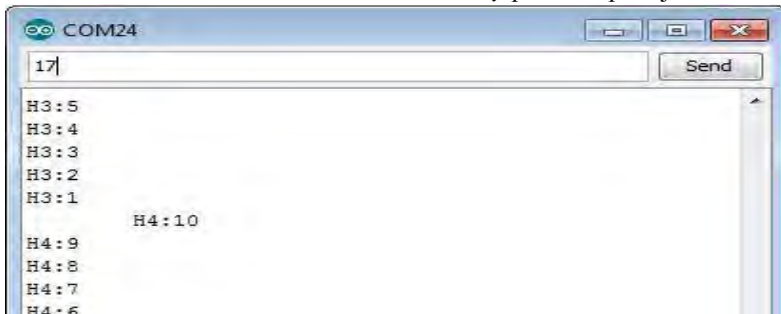
Gambar 4.8 Penambahan waktu *delay* pada lampu hijau 1



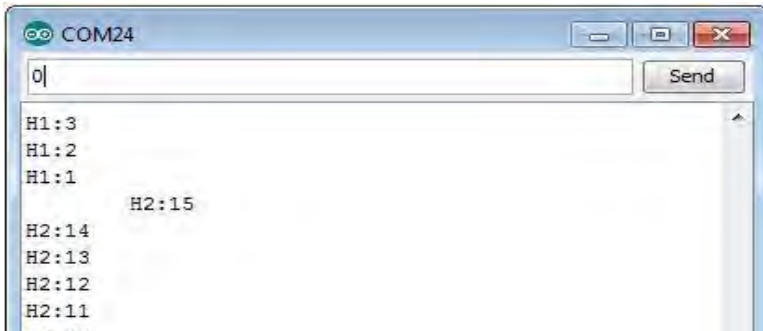
Gambar 4.9 Penambahan waktu *delay* pada lampu hijau 2



Gambar 4.10 Penambahan waktu *delay* pada lampu hijau 3



Gambar 4.11 Penambahan waktu *delay* pada lampu hijau 4



Gambar 4.12 Proses Reset Program ke kondisi semula

Dari pengujian program lampu lalu lintas dengan *serial monitor* yang kami peroleh dapat disimpulkan dari proses awal, penambahan waktu *delay* ke lampu 1, lampu 2, lampu 3, lampu 4, dan proses *reset* semua proses tersebut berjalan dengan baik karena besar waktu *delay* pada lampu sesuai dengan penambahan *input* waktu di android.

4.2.2 Pengujian Program *MIT App Inventor*

Perancangan program berikutnya adalah program *App Inventor* untuk memonitoring waktu *delay* lampu hijau dan mengontrolnya.. Hal ini bertujuan untuk memudahkan para petugas di bagian lalu lintas yaitu kepolisian dan dinas perhubungan agar dapat mengurangi kepadatan kendaraan di persimpangan lampu lalu lintas dan dapat mengurangi kecelakaan yang sering terjadi di persimpangan *Traffic light* . Dikatakan berhasil ketika program *app inventor* sukses untuk koneksi pairing dengan *Bluetooth* maka dapat dilanjutkan dengan memonitoring dan mengontrol *time delay* lampu hijau. Program *app inventor* dapat bekerja dengan baik. Saat peng-*input*-an, waktu *delay* sesuai dengan lama waktu lampu. Kami mengambil data untuk setiap lampunya dengan berbagai *input*-an. Berikut ini pada Tabel 4.8, 4.9, 4.10 dan 4.11 adalah Tabel hasil pengujian dari program *Traffic light* menggunakan *MIT App Inventor* :

Tabel 4.8 Pengujian dari Program App inventot pada lampu 1 dengan *delay* awal 10 detik

Pengujian	<i>Input</i>	<i>Output</i>	<i>Input</i>	<i>Output</i>	<i>Input</i>	<i>Output</i>	Keterangan
1	5	15	10	20	0	10	Berhasil

Pengujian	<i>Input</i>	<i>Output</i>	<i>Input</i>	<i>Output</i>	<i>Input</i>	<i>Output</i>	Keterangan
2	5	15	10	20	0	10	Berhasil
3	5	15	10	20	0	10	Berhasil
4	5	15	10	20	0	10	Berhasil
5	5	15	10	20	0	10	Berhasil
6	8	18	13	23	0	10	Berhasil
7	8	18	13	23	0	10	Berhasil
8	8	18	13	23	0	10	Berhasil
9	8	18	13	23	0	10	Berhasil
10	8	18	13	23	0	10	Berhasil

Tabel 4.9 Pengujian dari Program App inventot pada lampu 2 dengan *delay* awal 10 detik

Pengujian	<i>Input</i>	<i>Output</i>	<i>Input</i>	<i>Output</i>	<i>Input</i>	<i>Output</i>	Keterangan
1	5	15	10	20	0	10	Berhasil
2	5	15	10	20	0	10	Berhasil
3	5	15	10	20	0	10	Berhasil
4	5	15	10	20	0	10	Berhasil
5	5	15	10	20	0	10	Berhasil
6	8	18	13	23	0	10	Berhasil
7	8	18	13	23	0	10	Berhasil
8	8	18	13	23	0	10	Berhasil
9	8	18	13	23	0	10	Berhasil
10	8	18	13	23	0	10	Berhasil

Tabel 4.10 Pengujian dari Program App inventot pada lampu 3 dengan *delay* awal 10 detik

Pengujian	<i>Input</i>	<i>Output</i>	<i>Input</i>	<i>Output</i>	<i>Input</i>	<i>Output</i>	Keterangan
1	5	15	10	20	0	10	Berhasil
2	5	15	10	20	0	10	Berhasil
3	5	15	10	20	0	10	Berhasil
4	5	15	10	20	0	10	Berhasil
5	5	15	10	20	0	10	Berhasil
6	8	18	13	23	0	10	Berhasil
7	8	18	13	23	0	10	Berhasil
8	8	18	13	23	0	10	Berhasil
9	8	18	13	23	0	10	Berhasil

Pengujian	<i>Input</i>	<i>Output</i>	<i>Input</i>	<i>Output</i>	<i>Input</i>	<i>Output</i>	Keterangan
10	8	18	13	23	0	10	Berhasil

Tabel 4.11 Pengujian dari Program App inventot pada lampu 4 dengan *delay* awal 10 detik

Pengujian	<i>Input</i>	<i>Output</i>	<i>Input</i>	<i>Output</i>	<i>Input</i>	<i>Output</i>	Keterangan
1	5	15	10	20	0	10	Berhasil
2	5	15	10	20	0	10	Berhasil
3	5	15	10	20	0	10	Berhasil
4	5	15	10	20	0	10	Berhasil
5	5	15	10	20	0	10	Berhasil
6	8	18	13	23	0	10	Berhasil
7	8	18	13	23	0	10	Berhasil
8	8	18	13	23	0	10	Berhasil
9	8	18	13	23	0	10	Berhasil
10	8	18	13	23	0	10	Berhasil

4.3 Pengujian Alat Secara Keseluruhan

Pengujian alat secara keseluruhan dihasilkan beberapa kesimpulan yakni :

- Jika arduino pada logika 0 maka *output*nya bernilai 0V
- Jika arduino pada logika 1 maka *output*nya bernilai 4,98V
- Hasil maksimal dari pengujian Bluetooth adalah 13 meter, setelah 13 meter Bluetooth tidak dapat menerima dan mengirim data.
- Tegangan keluaran rata-rata pada *power supply* 5volt tanpa beban adalah 4,99V dengan persentase kesalahan pada rangkaian sebesar 0,2 %
- Tegangan keluaran rata-rata pada *power supply* 9volt dengan beban adalah 9 V dengan persentase kesalahan pada rangkaian sebesar 0 %
- Rata-rata kerja *relay* yang kami uji sesuai dengan output nyala lampu pada lay out *Traffic light*.
- Pada saat pengujian progam app inventor dengan *delay* 10 detik di lampu hijau pertama tingkat keberhasilan : 0% gagal dan 100% berhasil.

- Pada saat pengujian program app inventor dengan *delay* 10 detik di lampu hijau kedua tingkat keberhasilan 0% gagal dan 100% berhasil.
- Pada saat pengujian program app inventor dengan *delay* 10 detik di lampu hijau ketiga tingkat keberhasilan 0% gagal dan 100% berhasil.
- Pada saat pengujian program app inventor dengan *delay* 10 detik di lampu hijau keempat tingkat keberhasilan 0% gagal dan 100% berhasil

Dibawah ini adalah Gambar implementasi perancangan alat pengaturan dan monitoring lampu lalu lintas berbasis mikrokontroler yang dikendalikan dengan aplikasi di android secara nirkabel.



Gambar 4.13 Perancangan *layout* lampu lalu lintas

BAB V

PENUTUP

Bab penutup berisi tentang kesimpulan-kesimpulan yang didapatkan selama proses pembuatan Tugas Akhir ini beserta saran - saran untuk perbaikan dan pengembangannya.

Kesimpulan

Berdasarkan proses simulasi, implementasi, pengujian dan analisis sistem dapat ditarik kesimpulan sebagai berikut :

1. Jarak komunikasi *Bluetooth* yang dapat dilakukan untuk mengirim dan menerima data adalah sejauh 12 ± 1 meter.
2. Tidak ada Pergeseran waktu antara jam RTC dengan jam digital laptop. Sehingga RTC dapat menjadi acuan untuk time shedule lampu lalu lintas.
3. Error yang dihasil dari tegangan keluaran Power Supply 5volt sebesar 0,10 - 0,12 volt dan pada power supply 9 volt sebesar 0 volt.
4. Penambahan delay waktu dengan app inventor baik satuan maupun puluhan dapat diterima dengan baik dengan presentase error sebesar 0%

Saran

Saran yang dapat diajukan untuk penelitian lebih lanjut mengenai topic ini adalah :

1. Diusahakan dalam membuat aplikasi *App Inventor*, dengan desain pada layar *handphone* dilengkapi fitur-fitur yang lengkap. Agar aplikasi yang dibuat dapat mudah digunakan dan mempunyai nilai jual.
2. Diusahakan pada pengujian bluetooth selanjutnya jarak maksimal dapat lebih jauh dan frekuensinya lebih cepat. Agar waktu pengiriman data dan penerimaannya bisa lebih efisien.
3. Diusahakan pada pengujian selanjutnya sistem pengontrolan ditambahkan kamera agar dapat dimonitoring juga melalui *handphone*.

4. Diusahakan untuk pembuatan *power supply*, tegangan yang dihasilkan lebih stabil lagi agar *error* yang dihasilkan lebih kecil.

DAFTAR PUSTAKA

- [1]**Lampu Lalu Lintas** , URL:
http://id.wikipedia.org/wiki/Lampu_lalu_lintas 31 Mei 2015
- [2] Hardesty,Larry 2010. *The MIT roots of Google's new software*,
Emereo Publishing, Hal 33.
- [3] Haryanto, Agus 2014. *Android Fast Track :Belajar Membuat
Aplikasi Android Dengan Mudah dan Cepat*, Agus Haryanto
Hal 19.
- [4] Suyanto, Muhammad 2005. **Pengantar Teknologi Informasi
Untuk Bisnis**,Penerbit Andi
- [5] Suhata 2005. **VB Sebagai Pusat Kendali Peralatan Elektronik,
Elex Media Komputido**. Hal 137
- [6] Jubilee,Enterprise 2010. **Teknik Menghemat Baterai**, Pt Elex
Medi Komputido. Hal 137
- [7]**Pengertian Bluetooth**, URL :
http://id.wikipedia.org/wiki/Bluetooth, 13 Mein 2015
- [8] Malik,Ibnu 2009. **Aneka Proyek Mikrokontroler PLC 16F84A**,
Pt Elex Media Komputindo Kelompok Gramedia. Hal 1
- [9]**Rangkaian Relay** , URL:
http://www.rangkaianelektronika.org/rangkaian-relay.htm, 15 Mei
2015.
- [10]**Rangkaian Relay** , URL:
http://www.rangkaianelektronika.org/rangkaian-relay.htm, 15 Mei
2015.

Halaman ini sengaja dikosongkan

LAMPIRAN 1

LISTING PROGRAM

a. Program Penambahan waktu *delay* lampu lalu lintas

```
#include <Wire.h>
#include "RTClib.h"
```

```
RTC_DS1307 RTC;
```

```
int red_1=13; //pin untuk lampu merah 1
int yel_1=12; //pin untuk lampu kuning 1
int grn_1=11; //pin untuk lampu hijau 1
int red_2=10; //pin untuk lampu merah 2
int yel_2=9; //pin untuk lampu kuning 2
int grn_2=8; //pin untuk lampu hijau 2
int red_3=7; //pin untuk lampu merah 3
int yel_3=6; //pin untuk lampu kuning 3
int grn_3=5; //pin untuk lampu hijau 3
int red_4=4; //pin untuk lampu merah 4
int yel_4=3; //pin untuk lampu kuning 4
int grn_4=2; //pin untuk lampu hijau 4
```

```
int a; //variable untuk countdown lampu
int numerik1 = 0; //nilai awal dalam pengaturan waktu lampu hijau 1
int numerik2 = 0; //nilai awal dalam pengaturan waktu lampu hijau 2
int numerik3 = 0; //nilai awal dalam pengaturan waktu lampu hijau 3
int numerik4 = 0; //nilai awal dalam pengaturan waktu lampu hijau 4
```

```
// program untuk menjalankan penambahan waktu
```

```
void setup()
{
  Serial.begin(9600);
  Wire.begin();
  RTC.begin();

  digitalWrite(A2, HIGH);
  digitalWrite(A3, LOW);
}
```

```

pinMode(A3,OUTPUT);
pinMode(A2,OUTPUT);
pinMode(red_1,OUTPUT);
pinMode(yel_1,OUTPUT);
pinMode(grn_1,OUTPUT);
pinMode(red_2,OUTPUT);
pinMode(yel_2,OUTPUT);
pinMode(grn_2,OUTPUT);
pinMode(red_3,OUTPUT);
pinMode(yel_3,OUTPUT);
pinMode(grn_3,OUTPUT);
pinMode(red_4,OUTPUT);
pinMode(yel_4,OUTPUT);
pinMode(grn_4,OUTPUT);

if(! RTC.isrunning()) {
  Serial.println("RTC is NOT running!");
  // following line sets the RTC to the date & time this sketch was
compiled
  //RTC.adjust(DateTime(__DATE__, __TIME__));
}
//RTC.adjust(DateTime(2015, 7, 6, 23, 42, 0)); //digunakan untuk
mensetting waktu awal

//untuk plant 0 pada time schedule sebagai tahap persiapan lampu lalu
lintas setelah sistem mati.
for (int i=0;i<60;i++)
{
  digitalWrite(yel_1,HIGH);
  digitalWrite(yel_2,HIGH);
  digitalWrite(yel_3,HIGH);
  digitalWrite(yel_4,HIGH);
  delay(1000);
  digitalWrite(yel_1,LOW);
  digitalWrite(yel_2,LOW);
  digitalWrite(yel_3,LOW);
  digitalWrite(yel_4,LOW);
  delay(1000);
}

```

```
//untuk mengakhiri kondisi awal lampu lalu lintas ke time schedule  
lampu lalu lintas.
```

```
    if (Serial.read()=='b')  
    {  
        break;  
    }  
}  
}
```

```
void loop()
```

```
{  
    DateTime now = RTC.now();  
    delay(1000);
```

```
    // Program time schedule lampu hijau
```

```
    if (now.hour()>8 && now.hour()<11) //Plan 1  
    {  
        plant1();  
    }
```

```
    if (now.hour()>10 && now.hour()<14) //Plan 2  
    {  
        plant2();  
    }
```

```
    if (now.hour()>14 && now.hour()<19) //Plan 3  
    {  
        plant3();  
    }
```

```
    if (now.hour()>18 && now.hour()<21) //Plan 4  
    {  
        plant4();  
    }
```

```
    if (now.hour()>20 && now.hour()<22) //Plan 5  
    {
```



```

    plant5();
}

if (now.hour()>21 && now.hour()<5) //Plan 6
{
    plant6();
}

if (now.hour()>4 && now.hour()<9) //Plan 7
{
    plant7();
}
}

//untuk plant 1 pada time schedule lampu lalu lintas
void plant1()
{
    digitalWrite (red_2,HIGH);
    digitalWrite (red_3,HIGH);
    digitalWrite (red_4,HIGH);
    digitalWrite (grn_1,HIGH);
    delay_green1();
    digitalWrite (grn_1,LOW);
    digitalWrite (yel_1,HIGH);
    delay(1500);
    digitalWrite (yel_1,LOW);
    digitalWrite (red_1,HIGH);
    digitalWrite (red_2,LOW);
    digitalWrite (yel_2,HIGH);
    delay(1500);
    digitalWrite (yel_2,LOW);
    digitalWrite (grn_2,HIGH);
    delay_green2();
    digitalWrite (grn_2,LOW);
    digitalWrite (yel_2,HIGH);
    delay(1500);
    digitalWrite (yel_2,LOW);
    digitalWrite (red_2,HIGH);
    digitalWrite (red_3,LOW);

```

```

digitalWrite (yel_3,HIGH);
delay(1500);
digitalWrite (yel_3,LOW);
digitalWrite (grn_3,HIGH);
delay_green3();
digitalWrite (grn_3,LOW);
digitalWrite (yel_3,HIGH);
delay(1500);
digitalWrite (yel_3,LOW);
digitalWrite (red_3,HIGH);
digitalWrite (red_4,LOW);
digitalWrite (yel_4,HIGH);
delay(1500);
digitalWrite (yel_4,LOW);
digitalWrite (grn_4,HIGH);
delay_green4();
digitalWrite (grn_4,LOW);
digitalWrite (yel_4,HIGH);
delay(1500);
digitalWrite (yel_4,LOW);
digitalWrite (red_4,HIGH);
digitalWrite (red_1,LOW);
digitalWrite (yel_1,HIGH);
delay(1500);
digitalWrite (yel_1,LOW);
}

```

```

//untuk plant 2 pada time schedule lampu lalu lintas
void plant2()

```

```

{
    digitalWrite (red_2,HIGH);
    digitalWrite (red_3,HIGH);
    digitalWrite (red_4,HIGH);
    digitalWrite (grn_1,HIGH);
    delay_green1b();
    digitalWrite (grn_1,LOW);
    digitalWrite (yel_1,HIGH);
    delay(1500);
    digitalWrite (yel_1,LOW);
}

```

```

digitalWrite (red_1,HIGH);
digitalWrite (red_2,LOW);
digitalWrite (yel_2,HIGH);
delay(1500);
digitalWrite (yel_2,LOW);
digitalWrite (grn_2,HIGH);
delay_green2b();
digitalWrite (grn_2,LOW);
digitalWrite (yel_2,HIGH);
delay(1500);
digitalWrite (yel_2,LOW);
digitalWrite (red_2,HIGH);
digitalWrite (red_3,LOW);
digitalWrite (yel_3,HIGH);
delay(1500);
digitalWrite (yel_3,LOW);
digitalWrite (grn_3,HIGH);
delay_green3b();
digitalWrite (grn_3,LOW);
digitalWrite (yel_3,HIGH);
delay(1500);
digitalWrite (yel_3,LOW);
digitalWrite (red_3,HIGH);
digitalWrite (red_4,LOW);
digitalWrite (yel_4,HIGH);
delay(1500);
digitalWrite (yel_4,LOW);
digitalWrite (grn_4,HIGH);
delay_green4b();
digitalWrite (grn_4,LOW);
digitalWrite (yel_4,HIGH);
delay(1500);
digitalWrite (yel_4,LOW);
digitalWrite (red_4,HIGH);
digitalWrite (red_1,LOW);
digitalWrite (yel_1,HIGH);
delay(1500);
digitalWrite (yel_1,LOW);
}

```

```

//untuk plant 3 pada time schedule lampu lalu lintas
void plant3()
{
    digitalWrite (red_2,HIGH);
    digitalWrite (red_3,HIGH);
    digitalWrite (red_4,HIGH);
    digitalWrite (grn_1,HIGH);
    delay_green1c();
    digitalWrite (grn_1,LOW);
    digitalWrite (yel_1,HIGH);
    delay(1500);
    digitalWrite (yel_1,LOW);
    digitalWrite (red_1,HIGH);
    digitalWrite (red_2,LOW);
    digitalWrite (yel_2,HIGH);
    delay(1500);
    digitalWrite (yel_2,LOW);
    digitalWrite (grn_2,HIGH);
    delay_green2c();
    digitalWrite (grn_2,LOW);
    digitalWrite (yel_2,HIGH);
    delay(1500);
    digitalWrite (yel_2,LOW);
    digitalWrite (red_2,HIGH);
    digitalWrite (red_3,LOW);
    digitalWrite (yel_3,HIGH);
    delay(1500);
    digitalWrite (yel_3,LOW);
    digitalWrite (grn_3,HIGH);
    delay_green3c();
    digitalWrite (grn_3,LOW);
    digitalWrite (yel_3,HIGH);
    delay(1500);
    digitalWrite (yel_3,LOW);
    digitalWrite (red_3,HIGH);
    digitalWrite (red_4,LOW);
    digitalWrite (yel_4,HIGH);
    delay(1500);
}

```

```

digitalWrite (yel_4,LOW);
digitalWrite (grn_4,HIGH);
delay_green4c();
digitalWrite (grn_4,LOW);
digitalWrite (yel_4,HIGH);
delay(1500);
digitalWrite (yel_4,LOW);
digitalWrite (red_4,HIGH);
digitalWrite (red_1,LOW);
digitalWrite (yel_1,HIGH);
delay(1500);
digitalWrite (yel_1,LOW);
}

```

//untuk plant 4 pada time schedule lampu lalu lintas

```

void plant4()
{
    digitalWrite (red_2,HIGH);
    digitalWrite (red_3,HIGH);
    digitalWrite (red_4,HIGH);
    digitalWrite (grn_1,HIGH);
    delay_green1d();
    digitalWrite (grn_1,LOW);
    digitalWrite (yel_1,HIGH);
    delay(1500);
    digitalWrite (yel_1,LOW);
    digitalWrite (red_1,HIGH);
    digitalWrite (red_2,LOW);
    digitalWrite (yel_2,HIGH);
    delay(1500);
    digitalWrite (yel_2,LOW);
    digitalWrite (grn_2,HIGH);
    delay_green2d();
    digitalWrite (grn_2,LOW);
    digitalWrite (yel_2,HIGH);
    delay(1500);
    digitalWrite (yel_2,LOW);
    digitalWrite (red_2,HIGH);
    digitalWrite (red_3,LOW);
}

```

```

digitalWrite (yel_3,HIGH);
delay(1500);
digitalWrite (yel_3,LOW);
digitalWrite (grn_3,HIGH);
delay_green3d();
digitalWrite (grn_3,LOW);
digitalWrite (yel_3,HIGH);
delay(1500);
digitalWrite (yel_3,LOW);
digitalWrite (red_3,HIGH);
digitalWrite (red_4,LOW);
digitalWrite (yel_4,HIGH);
delay(1500);
digitalWrite (yel_4,LOW);
digitalWrite (grn_4,HIGH);
delay_green4d();
digitalWrite (grn_4,LOW);
digitalWrite (yel_4,HIGH);
delay(1500);
digitalWrite (yel_4,LOW);
digitalWrite (red_4,HIGH);
digitalWrite (red_1,LOW);
digitalWrite (yel_1,HIGH);
delay(1500);
digitalWrite (yel_1,LOW);
}

```

```

//untuk plant 5 pada time schedule lampu lalu lintas
void plant5()

```

```

{
digitalWrite (red_2,HIGH);
digitalWrite (red_3,HIGH);
digitalWrite (red_4,HIGH);
digitalWrite (grn_1,HIGH);
delay_green1e();
digitalWrite (grn_1,LOW);
digitalWrite (yel_1,HIGH);
delay(1500);
digitalWrite (yel_1,LOW);
}

```

```

digitalWrite (red_1,HIGH);
digitalWrite (red_2,LOW);
digitalWrite (yel_2,HIGH);
delay(1500);
digitalWrite (yel_2,LOW);
digitalWrite (grn_2,HIGH);
delay_green2e();
digitalWrite (grn_2,LOW);
digitalWrite (yel_2,HIGH);
delay(1500);
digitalWrite (yel_2,LOW);
digitalWrite (red_2,HIGH);
digitalWrite (red_3,LOW);
digitalWrite (yel_3,HIGH);
delay(1500);
digitalWrite (yel_3,LOW);
digitalWrite (grn_3,HIGH);
delay_green3e();
digitalWrite (grn_3,LOW);
digitalWrite (yel_3,HIGH);
delay(1500);
digitalWrite (yel_3,LOW);
digitalWrite (red_3,HIGH);
digitalWrite (red_4,LOW);
digitalWrite (yel_4,HIGH);
delay(1500);
digitalWrite (yel_4,LOW);
digitalWrite (grn_4,HIGH);
delay_green4e();
digitalWrite (grn_4,LOW);
digitalWrite (yel_4,HIGH);
delay(1500);
digitalWrite (yel_4,LOW);
digitalWrite (red_4,HIGH);
digitalWrite (red_1,LOW);
digitalWrite (yel_1,HIGH);
delay(1500);
digitalWrite (yel_1,LOW);
}

```

```

//untuk plant 6 pada time schedule lampu lalu lintas
void plant6()
{
    digitalWrite (red_2,HIGH);
    digitalWrite (red_3,HIGH);
    digitalWrite (red_4,HIGH);
    digitalWrite (grn_1,HIGH);
    delay_green1f();
    digitalWrite (grn_1,LOW);
    digitalWrite (yel_1,HIGH);
    delay(1500);
    digitalWrite (yel_1,LOW);
    digitalWrite (red_1,HIGH);
    digitalWrite (red_2,LOW);
    digitalWrite (yel_2,HIGH);
    delay(1500);
    digitalWrite (yel_2,LOW);
    digitalWrite (grn_2,HIGH);
    delay_green2f();
    digitalWrite (grn_2,LOW);
    digitalWrite (yel_2,HIGH);
    delay(1500);
    digitalWrite (yel_2,LOW);
    digitalWrite (red_2,HIGH);
    digitalWrite (red_3,LOW);
    digitalWrite (yel_3,HIGH);
    delay(1500);
    digitalWrite (yel_3,LOW);
    digitalWrite (grn_3,HIGH);
    delay_green3f();
    digitalWrite (grn_3,LOW);
    digitalWrite (yel_3,HIGH);
    delay(1500);
    digitalWrite (yel_3,LOW);
    digitalWrite (red_3,HIGH);
    digitalWrite (red_4,LOW);
    digitalWrite (yel_4,HIGH);

```



```

delay(1500);
digitalWrite (yel_4,LOW);
digitalWrite (grn_4,HIGH);
delay_green4f();
digitalWrite (grn_4,LOW);
digitalWrite (yel_4,HIGH);
delay(1500);
digitalWrite (yel_4,LOW);
digitalWrite (red_4,HIGH);
digitalWrite (red_1,LOW);
digitalWrite (yel_1,HIGH);
delay(1500);
digitalWrite (yel_1,LOW);
}

```

//untuk plant 7 pada time schedule lampu lalu lintas
void plant7()

```

{
    digitalWrite (red_2,HIGH);
    digitalWrite (red_3,HIGH);
    digitalWrite (red_4,HIGH);
    digitalWrite (grn_1,HIGH);
    delay_green1g();
    digitalWrite (grn_1,LOW);
    digitalWrite (yel_1,HIGH);
    delay(1500);
    digitalWrite (yel_1,LOW);
    digitalWrite (red_1,HIGH);
    digitalWrite (red_2,LOW);
    digitalWrite (yel_2,HIGH);
    delay(1500);
    digitalWrite (yel_2,LOW);
    digitalWrite (grn_2,HIGH);
    delay_green2g();
    digitalWrite (grn_2,LOW);
    digitalWrite (yel_2,HIGH);
    delay(1500);
    digitalWrite (yel_2,LOW);
}

```

```

digitalWrite (red_2,HIGH);
digitalWrite (red_3,LOW);
digitalWrite (yel_3,HIGH);
delay(1500);
digitalWrite (yel_3,LOW);
digitalWrite (grn_3,HIGH);
delay_green3g();
digitalWrite (grn_3,LOW);
digitalWrite (yel_3,HIGH);
delay(1500);
digitalWrite (yel_3,LOW);
digitalWrite (red_3,HIGH);
digitalWrite (red_4,LOW);
digitalWrite (yel_4,HIGH);
delay(1500);
digitalWrite (yel_4,LOW);
digitalWrite (grn_4,HIGH);
delay_green4g();
digitalWrite (grn_4,LOW);
digitalWrite (yel_4,HIGH);
delay(1500);
digitalWrite (yel_4,LOW);
digitalWrite (red_4,HIGH);
digitalWrite (red_1,LOW);
digitalWrite (yel_1,HIGH);
delay(1500);
digitalWrite (yel_1,LOW);
}

```

//Pengaturan input angka delay waktu lampu hijau 1,2,3,dan 4

```

void input1() //untuk input angka pada lampu hijau 1
{
  if(Serial.available())
  {
    numerik1=0; // untuk mereset nilai delay sebelumnya
    int BlueData = Serial.read();//untuk menyimpan nilai pada serial ke
    BlueData
  }
}

```

```

//untuk mengubah input string menjadi integer
while (BlueData != '\n')
{
    if (isdigit(BlueData))
    {
        numerik1 = numerik1*10 + (BlueData - '0');
    }
    BlueData = Serial.read();
}

//untuk range nilai input angka pada lampu hijau 1
numerik1 = constrain(numerik1, 0, 60);

//untuk menjalankan fungsi pengaturan ketika ada input delay pada
lampu hijau 1
if (BlueData == '\n')
{
    gantidelay = true;
}

Serial.print ("numerik1 :");
Serial.println (numerik1);

}
}

void input2() //untuk input angka pada lampu hijau 2
{
    if(Serial.available())
    {
        numerik2=0;
        int BlueData = Serial.read();

        while (BlueData != '\n')
        {
            if (isdigit(BlueData))
            {
                numerik2 = numerik2*10 + (BlueData - '0');
            }
        }
    }
}

```

```

    BlueData = Serial.read();
}

//untuk range nilai input angka pada lampu hijau 2
numerik2 = constrain(numerik2, 0, 60);

//untuk menjalankan fungsi pengaturan ketika ada input delay pada
lampu hijau 2
if (BlueData == '\n')
{
    gantidelay = true;
}

Serial.print ("numerik2 :");
Serial.println (numerik2);

}
}

void input3() //untuk input angka pada lampu hijau 3
{
    if(Serial.available())
    {
        numerik3=0;
        int BlueData = Serial.read();

        while (BlueData != '\n')
        {
            if (isdigit(BlueData))
            {
                numerik3 = numerik3*10 + (BlueData - '0');
            }
            BlueData = Serial.read();
        }

        //untuk range nilai input angka pada lampu hijau 3
        numerik3 = constrain(numerik3, 0, 60);
    }
}

```

```

//untuk menjalankan fungsi pengaturan ketika ada input delay pada
lampu hijau 3
if (BlueData == '\n')
{
    gantidelay = true;
}

Serial.print ("numerik3 :");
Serial.println (numerik3);

}
}

void input4() //untuk input angka pada lampu hijau 4
{
    if(Serial.available())
    {
        numerik4=0;
        int BlueData = Serial.read();

        while (BlueData != '\n')
        {
            if (isdigit(BlueData))
            {
                numerik4 = numerik4*10 + (BlueData - '0');
            }
            BlueData = Serial.read();
        }

        //untuk range nilai input angka pada lampu hijau 4
        numerik4 = constrain(numerik4, 0, 60);

        //untuk menjalankan fungsi pengaturan ketika ada input delay pada
        lampu hijau 4
        if (BlueData == '\n')
        {
            gantidelay = true;
        }
    }
}

```

```

    Serial.print ("numerik4 :");
    Serial.println (numerik4);

}
}

//program countdown lampu hijau plan 1

void delay_green1() //lampu hijau ke 1 pada plan 1
{
    //untuk mengatur nilai delay lampu hijau ke 1, baik mengurangi
    maupun menambahkan.
    if (gantidelay){a=numerik1+1;}
    else {a=20;}

    while (a!=1)
    {
        delay (1000);
        a--;
        Serial.print ("H1:");
        Serial.println (a);
    }

    while (Serial.available()>0)
    {
        char pilihlampu=Serial.read();
        switch (pilihlampu)
        {
            case '1' : input1(); break;
            case '2' : input2(); break;
            case '3' : input3(); break;
            case '4' : input4(); break;
        }
    }

    Serial.print ("\t");
}

void delay_green2() //lampu hijau ke 2 pada plan 1

```

```

{
  //untuk mengatur nilai delay lampu hijau ke 2, baik mengurangi
  maupun menambahkan.
  if (gantidelay){a=numerik2+1;}
  else {a=41;}

  while (a!=1)
  {
    delay (1000);
    a--;
    Serial.print ("H2:");
    Serial.println (a);
  }

  while (Serial.available()>0)
  {
    char pilihlampu=Serial.read();
    switch (pilihlampu)
    {
      case '1' : input1(); break;
      case '2' : input2(); break;
      case '3' : input3(); break;
      case '4' : input4(); break;
    }
  }

  Serial.print ("\t");
}

void delay_green3() //lampu hijau ke 3 pada plan 1
{
  //untuk mengatur nilai delay lampu hijau ke 3, baik mengurangi
  maupun menambahkan.
  if (gantidelay){a=numerik3+1;}
  else {a=34;}

  while (a!=1)
  {
    delay (1000);

```

```

a--;
Serial.print ("H3:");
Serial.println (a);
}

while (Serial.available()>0)
{
char pilihlampu=Serial.read();
switch (pilihlampu)
{
case '1' : input1(); break;
case '2' : input2(); break;
case '3' : input3(); break;
case '4' : input4(); break;
}
}

Serial.print ("\t");
}

void delay_green4() //lampu hijau ke 4 pada plan 1
{
//untuk mengatur nilai delay lampu hijau ke 4, baik mengurangi
maupun menambahkan.
if (gantidelay){a=numerik4+1;}
else {a=26;}

while (a!=1)
{
delay (1000);
a--;
Serial.print ("H4:");
Serial.println (a);
}

while (Serial.available()>0)
{
char pilihlampu=Serial.read();
switch (pilihlampu)

```



```

    {
    case '1' : input1(); break;
    case '2' : input2(); break;
    case '3' : input3(); break;
    case '4' : input4(); break;
    }
}

Serial.print ("\t");
}

//program countdown lampu hijau plan 2

void delay_green1b() //lampu hijau ke 1 pada plan 2
{
    //untuk mengatur nilai delay lampu hijau ke 1, baik mengurangi
    maupun menambahkan.
    if (gantidelay){a=numerik1+1;}
    else {a=26;}

    while (a!=1)
    {
        delay (1000);
        a--;
        Serial.print ("H1:");
        Serial.println (a);
    }

    while (Serial.available(>0)
    {
        char pilihlampu=Serial.read();
        switch (pilihlampu)
        {
            case '1' : input1(); break;
            case '2' : input2(); break;
            case '3' : input3(); break;
            case '4' : input4(); break;
        }
    }
}

```

```
Serial.print ("\t");  
}
```

```
void delay_green2b() //lampu hijau ke 2 pada plan 2  
{  
    //untuk mengatur nilai delay lampu hijau ke 2, baik mengurangi  
    maupun menambahkan.  
    if (gantidelay){a=numerik2+1;}  
    else {a=41;}  
}
```

```
while (a!=1)  
{  
    delay (1000);  
    a--;  
    Serial.print ("H2:");  
    Serial.println (a);  
}
```

```
while (Serial.available()>0)  
{  
    char pilihlampu=Serial.read();  
    switch (pilihlampu)  
    {  
        case '1' : input1(); break;  
        case '2' : input2(); break;  
        case '3' : input3(); break;  
        case '4' : input4(); break;  
    }  
}
```

```
Serial.print ("\t");  
}
```

```
void delay_green3b() //lampu hijau ke 3 pada plan 2  
{  
    //untuk mengatur nilai delay lampu hijau ke 3, baik mengurangi  
    maupun menambahkan.  
    if (gantidelay){a=numerik3+1;}  
}
```

```

else {a=34;}

while (a!=1)
{
    delay (1000);
    a--;
    Serial.print ("H3:");
    Serial.println (a);
}

while (Serial.available()>0)
{
    char pilihlampu=Serial.read();
    switch (pilihlampu)
    {
        case '1' : input1(); break;
        case '2' : input2(); break;
        case '3' : input3(); break;
        case '4' : input4(); break;
    }
}

Serial.print ("\t");
}

void delay_green4b() //lampu hijau ke 4 pada plan 2
{
    //untuk mengatur nilai delay lampu hijau ke 4, baik mengurangi
    maupun menambahkan.
    if (gantidelay){a=numerik4+1;}
    else {a=26;}

    while (a!=1)
    {
        delay (1000);
        a--;
        Serial.print ("H4:");
        Serial.println (a);
    }
}

```

```

while (Serial.available()>0)
{
  char pilihlampu=Serial.read();
  switch (pilihlampu)
  {
    case '1' : input1(); break;
    case '2' : input2(); break;
    case '3' : input3(); break;
    case '4' : input4(); break;
  }
}

Serial.print ("\t");
}

```

//program countdown lampu hijau plan 3

```

void delay_green1c() //lampu hijau ke 1 pada plan 3
{
  //untuk mengatur nilai delay lampu hijau ke 1, baik mengurangi
  maupun menambahkan.
  if (gantidelay){a=numerik1+1;}
  else {a=19;}

  while (a!=1)
  {
    delay (1000);
    a--;
    Serial.print ("H1:");
    Serial.println (a);
  }
}

```

```

while (Serial.available()>0)
{
  char pilihlampu=Serial.read();
  switch (pilihlampu)
  {
    case '1' : input1(); break;

```

```

        case '2' : input2(); break;
        case '3' : input3(); break;
        case '4' : input4(); break;
    }
}

Serial.print ("\t");
}

void delay_green2c() //lampu hijau ke 2 pada plan 3
{
    //untuk mengatur nilai delay lampu hijau ke 2, baik mengurangi
    maupun menambahkan.
    if (gantidelay){a=numerik2+1;}
    else {a=27;}

    while (a!=1)
    {
        delay (1000);
        a--;
        Serial.print ("H2:");
        Serial.println (a);
    }

    while (Serial.available()>0)
    {
        char pilihlampu=Serial.read();
        switch (pilihlampu)
        {
            case '1' : input1(); break;
            case '2' : input2(); break;
            case '3' : input3(); break;
            case '4' : input4(); break;
        }
    }

    Serial.print ("\t");
}

```

```

void delay_green3c() //lampu hijau ke 3 pada plan 3
{
    //untuk mengatur nilai delay lampu hijau ke 3, baik mengurangi
    maupun menambahkan.
    if (gantidelay){a=numerik3+1;}
    else {a=34;}

    while (a!=1)
    {
        delay (1000);
        a--;
        Serial.print ("H3:");
        Serial.println (a);
    }

    while (Serial.available()>0)
    {
        char pilihlampu=Serial.read();
        switch (pilihlampu)
        {
            case '1' : input1(); break;
            case '2' : input2(); break;
            case '3' : input3(); break;
            case '4' : input4(); break;
        }
    }

    Serial.print ("\t");
}

void delay_green4c() //lampu hijau ke 4 pada plan 3
{
    //untuk mengatur nilai delay lampu hijau ke 4, baik mengurangi
    maupun menambahkan.
    if (gantidelay){a=numerik4+1;}
    else {a=26;}

    while (a!=1)
    {

```

```

    delay (1000);
    a--;
    Serial.print ("H4:");
    Serial.println (a);
}

while (Serial.available()>0)
{
    char pilihlampu=Serial.read();
    switch (pilihlampu)
    {
        case '1' : input1(); break;
        case '2' : input2(); break;
        case '3' : input3(); break;
        case '4' : input4(); break;
    }
}

Serial.print ("\t");
}

//program countdown lampu hijau plan 4

void delay_green1d() //lampu hijau ke 1 pada plan 4
{
    //untuk mengatur nilai delay lampu hijau ke 1, baik mengurangi
    maupun menambahkan.
    if (gantidelay){a=numerik1+1;}
    else {a=19;}

    while (a!=1)
    {
        delay (1000);
        a--;
        Serial.print ("H1:");
        Serial.println (a);
    }

    while (Serial.available()>0)

```

```

{
char pilihlampu=Serial.read();
switch (pilihlampu)
{
case '1' : input1(); break;
case '2' : input2(); break;
case '3' : input3(); break;
case '4' : input4(); break;
}
}

Serial.print ("\t");
}

```

```

void delay_green2d() //lampu hijau ke 2 pada plan 4
{
//untuk mengatur nilai delay lampu hijau ke 2, baik mengurangi
maupun menambahkan.
if (gantidelay){a=numerik2+1;}
else {a=39;}

```

```

while (a!=1)
{
delay (1000);
a--;
Serial.print ("H2:");
Serial.println (a);
}

```

```

while (Serial.available()>0)
{
char pilihlampu=Serial.read();
switch (pilihlampu)
{
case '1' : input1(); break;
case '2' : input2(); break;
case '3' : input3(); break;
case '4' : input4(); break;
}
}

```



```

    }

    Serial.print ("\t");
}

void delay_green3d() //lampu hijau ke 3 pada plan 4
{
    //untuk mengatur nilai delay lampu hijau ke 3, baik mengurangi
    maupun menambahkan.
    if (gantidelay){a=numerik3+1;}
    else {a=37;}

    while (a!=1)
    {
        delay (1000);
        a--;
        digitalWrite (grn_3,HIGH);
        digitalWrite (red_4,HIGH);
        Serial.print ("H3:");
        Serial.println (a);
        //Serial.print ("+" :");
        //Serial.println (numerik3);
    }
    Serial.print ("\t");
    input3();
}

void delay_green4d() //lampu hijau ke 4 pada plan 4
{
    //untuk mengatur nilai delay lampu hijau ke 4, baik mengurangi
    maupun menambahkan.
    if (gantidelay){a=numerik4+1;}
    else {a=36;}

    while (a!=1)
    {
        delay (1000);
        a--;
        Serial.print ("H4:");
    }
}

```

```

    Serial.println (a);
}

while (Serial.available()>0)
{
    char pilihlampu=Serial.read();
    switch (pilihlampu)
    {
        case '1' : input1(); break;
        case '2' : input2(); break;
        case '3' : input3(); break;
        case '4' : input4(); break;
    }
}

Serial.print ("\t");
}

//program countdown lampu hijau plan 5

void delay_green1e() //lampu hijau ke 1 pada plan 5
{
    //untuk mengatur nilai delay lampu hijau ke 1, baik mengurangi
    maupun menambahkan.
    if (gantidelay){a=numerik1+1;}
    else {a=19;}

    while (a!=1)
    {
        delay (1000);
        a--;
        Serial.print ("H1:");
        Serial.println (a);
    }

    while (Serial.available()>0)
    {
        char pilihlampu=Serial.read();
        switch (pilihlampu)

```

```

    {
    case '1' : input1(); break;
    case '2' : input2(); break;
    case '3' : input3(); break;
    case '4' : input4(); break;
    }
}

Serial.print ("\t");
}

void delay_green2e() //lampu hijau ke 2 pada plan 5
{
    //untuk mengatur nilai delay lampu hijau ke 2, baik mengurangi
    maupun menambahkan.
    if (gantidelay){a=numerik2+1;}
    else {a=31;}

    while (a!=1)
    {
        delay (1000);
        a--;
        Serial.print ("H2:");
        Serial.println (a);
    }

    while (Serial.available()>0)
    {
        char pilihlampu=Serial.read();
        switch (pilihlampu)
        {
            case '1' : input1(); break;
            case '2' : input2(); break;
            case '3' : input3(); break;
            case '4' : input4(); break;
        }
    }

    Serial.print ("\t");

```

```

}

void delay_green3e() //lampu hijau ke 3 pada plan 5
{
    //untuk mengatur nilai delay lampu hijau ke 3, baik mengurangi
    maupun menambahkan.
    if (gantidelay){a=numerik3+1;}
    else {a=41;}

    while (a!=1)
    {
        delay (1000);
        a--;
        Serial.print ("H3:");
        Serial.println (a);
    }

    while (Serial.available()>0)
    {
        char pilihlampu=Serial.read();
        switch (pilihlampu)
        {
            case '1' : input1(); break;
            case '2' : input2(); break;
            case '3' : input3(); break;
            case '4' : input4(); break;
        }
    }

    Serial.print ("\t");
}

void delay_green4e() //lampu hijau ke 4 pada plan 5
{
    //untuk mengatur nilai delay lampu hijau ke 4, baik mengurangi
    maupun menambahkan.
    if (gantidelay){a=numerik4+1;}
    else {a=26;}

```

```

while (a!=1)
{
    delay (1000);
    a--;
    Serial.print ("H4:");
    Serial.println (a);
}

while (Serial.available()>0)
{
    char pilihlampu=Serial.read();
    switch (pilihlampu)
    {
        case '1' : input1(); break;
        case '2' : input2(); break;
        case '3' : input3(); break;
        case '4' : input4(); break;
    }
}

Serial.print ("\t");
}

//program countdown lampu hijau plan 6

void delay_green1f() //lampu hijau ke 1 pada plan 6
{
    //untuk mengatur nilai delay lampu hijau ke 1, baik mengurangi
    maupun menambahkan.
    if (gantidelay){a=numerik1+1;}
    else {a=16;}

    while (a!=1)
    {
        delay (1000);
        a--;
        Serial.print ("H1:");
        Serial.println (a);
    }
}

```

```

while (Serial.available()>0)
{
  char pilihlampu=Serial.read();
  switch (pilihlampu)
  {
    case '1' : input1(); break;
    case '2' : input2(); break;
    case '3' : input3(); break;
    case '4' : input4(); break;
  }
}

Serial.print ("\t");
}

void delay_green2f() //lampu hijau ke 2 pada plan 6
{
  //untuk mengatur nilai delay lampu hijau ke 2, baik mengurangi
  maupun menambahkan.
  if (gantidelay){a=numerik2+1;}
  else {a=21;}

  while (a!=1)
  {
    delay (1000);
    a--;
    Serial.print ("H2:");
    Serial.println (a);
  }

  while (Serial.available()>0)
  {
    char pilihlampu=Serial.read();
    switch (pilihlampu)
    {
      case '1' : input1(); break;
      case '2' : input2(); break;
      case '3' : input3(); break;
    }
  }
}

```

```

        case '4' : input4(); break;
    }
}

Serial.print ("\t");
}

void delay_green3f() //lampu hijau ke 3 pada plan 6
{
    //untuk mengatur nilai delay lampu hijau ke 3, baik mengurangi
    maupun menambahkan.
    if (gantidelay){a=numerik3+1;}
    else {a=11;}

    while (a!=1)
    {
        delay (1000);
        a--;
        Serial.print ("H3:");
        Serial.println (a);
    }

    while (Serial.available()>0)
    {
        char pilihlampu=Serial.read();
        switch (pilihlampu)
        {
            case '1' : input1(); break;
            case '2' : input2(); break;
            case '3' : input3(); break;
            case '4' : input4(); break;
        }
    }

    Serial.print ("\t");
}

void delay_green4f() //lampu hijau ke 4 pada plan 6
{

```

//untuk mengatur nilai delay lampu hijau ke 4, baik mengurangi maupun menambahkan.

```
if (gantidelay){a=numerik4+1;}  
else {a=16;}
```

```
while (a!=1)  
{  
    delay (1000);  
    a--;  
    Serial.print ("H4:");  
    Serial.println (a);  
}
```

```
while (Serial.available()>0)  
{  
    char pilihlampu=Serial.read();  
    switch (pilihlampu)  
    {  
        case '1' : input1(); break;  
        case '2' : input2(); break;  
        case '3' : input3(); break;  
        case '4' : input4(); break;  
    }  
}
```

```
Serial.print ("\t");  
}
```

//program countdown lampu hijau plan 7

```
void delay_green1g() //lampu hijau ke 1 pada plan 7  
{
```

//untuk mengatur nilai delay lampu hijau ke 1, baik mengurangi maupun menambahkan.

```
if (gantidelay){a=numerik1+1;}  
else {a=16;}
```

```
while (a!=1)  
{
```



```

    delay (1000);
    a--;
    Serial.print ("H1:");
    Serial.println (a);
}

while (Serial.available()>0)
{
    char pilihlampu=Serial.read();
    switch (pilihlampu)
    {
        case '1' : input1(); break;
        case '2' : input2(); break;
        case '3' : input3(); break;
        case '4' : input4(); break;
    }
}

Serial.print ("\t");
}

void delay_green2g() //lampu hijau ke 2 pada plan 7
{
    //untuk mengatur nilai delay lampu hijau ke 2, baik mengurangi
    maupun menambahkan.
    if (gantidelay){a=numerik2+1;}
    else {a=21;}

    while (a!=1)
    {
        delay (1000);
        a--;
        Serial.print ("H2:");
        Serial.println (a);
    }

    while (Serial.available()>0)
    {
        char pilihlampu=Serial.read();

```

```

switch (pilihlampu)
{
    case '1' : input1(); break;
    case '2' : input2(); break;
    case '3' : input3(); break;
    case '4' : input4(); break;
}
}

Serial.print ("t");
}

void delay_green3g() //lampu hijau ke 3 pada plan 7
{
    //untuk mengatur nilai delay lampu hijau ke 3, baik mengurangi
    maupun menambahkan.
    if (gantidelay){a=numerik3+1;}
    else {a=31;}

    while (a!=1)
    {
        delay (1000);
        a--;
        Serial.print ("H3:");
        Serial.println (a);
    }

    while (Serial.available()>0)
    {
        char pilihlampu=Serial.read();
        switch (pilihlampu)
        {
            case '1' : input1(); break;
            case '2' : input2(); break;
            case '3' : input3(); break;
            case '4' : input4(); break;
        }
    }
}

```

```

Serial.print ("\t");
}

void delay_green4g() //lampu hijau ke 4 pada plan 7
{
    //untuk mengatur nilai delay lampu hijau ke 4, baik mengurangi
    maupun menambahkan.
    if (gantidelay){a=numerik4+1;}
    else {a=26;}

    while (a!=1)
    {
        delay (1000);
        a--;
        Serial.print ("H4:");
        Serial.println (a);
    }

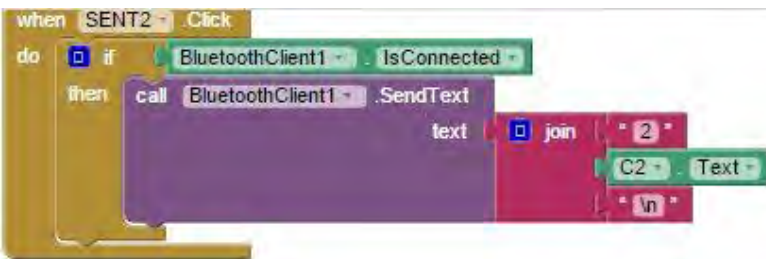
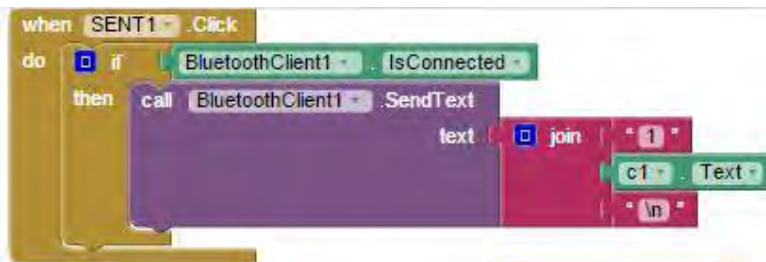
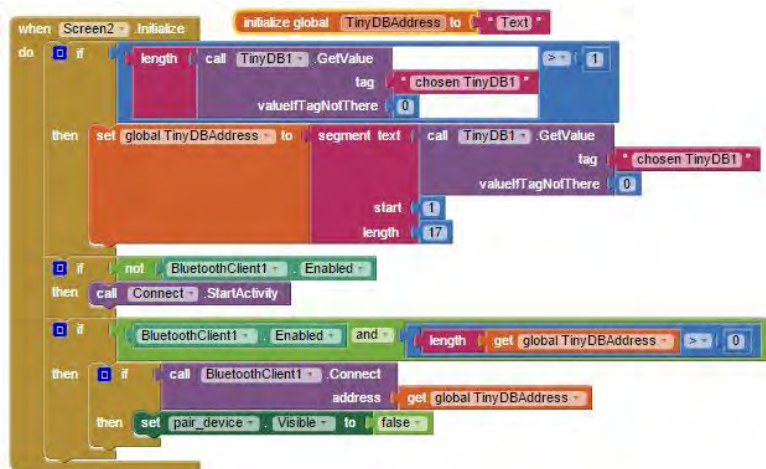
    while (Serial.available()>0)
    {
        char pilihlampu=Serial.read();
        switch (pilihlampu)
        {
            case '1' : input1(); break;
            case '2' : input2(); break;
            case '3' : input3(); break;
            case '4' : input4(); break;
        }
    }

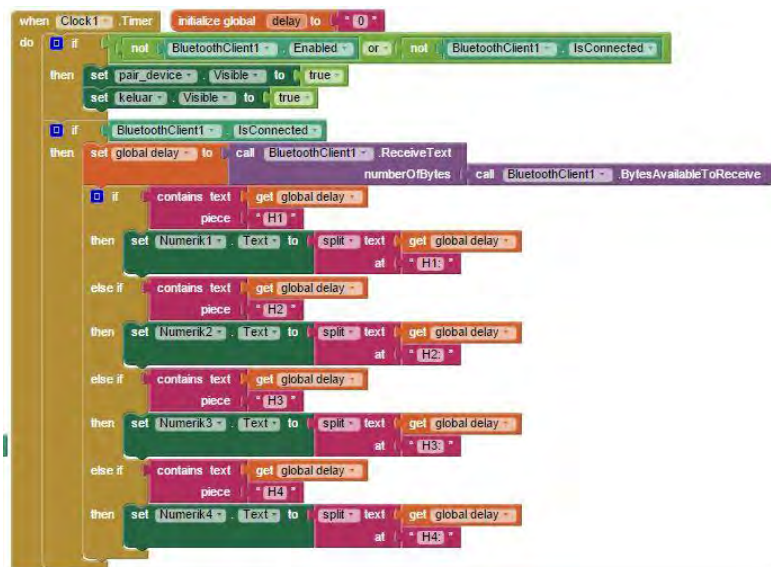
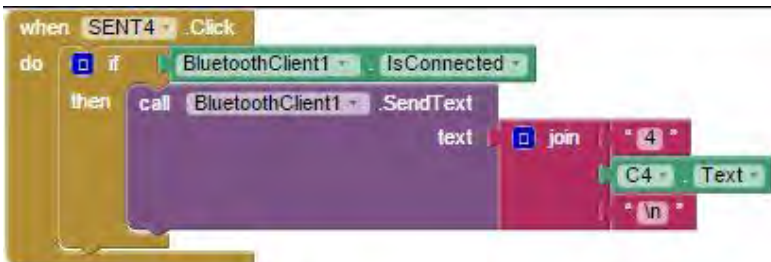
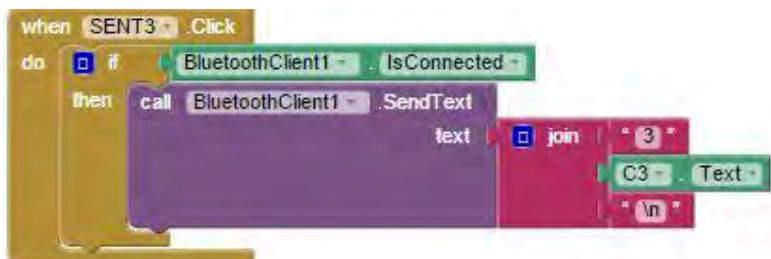
    Serial.print ("\t");
}

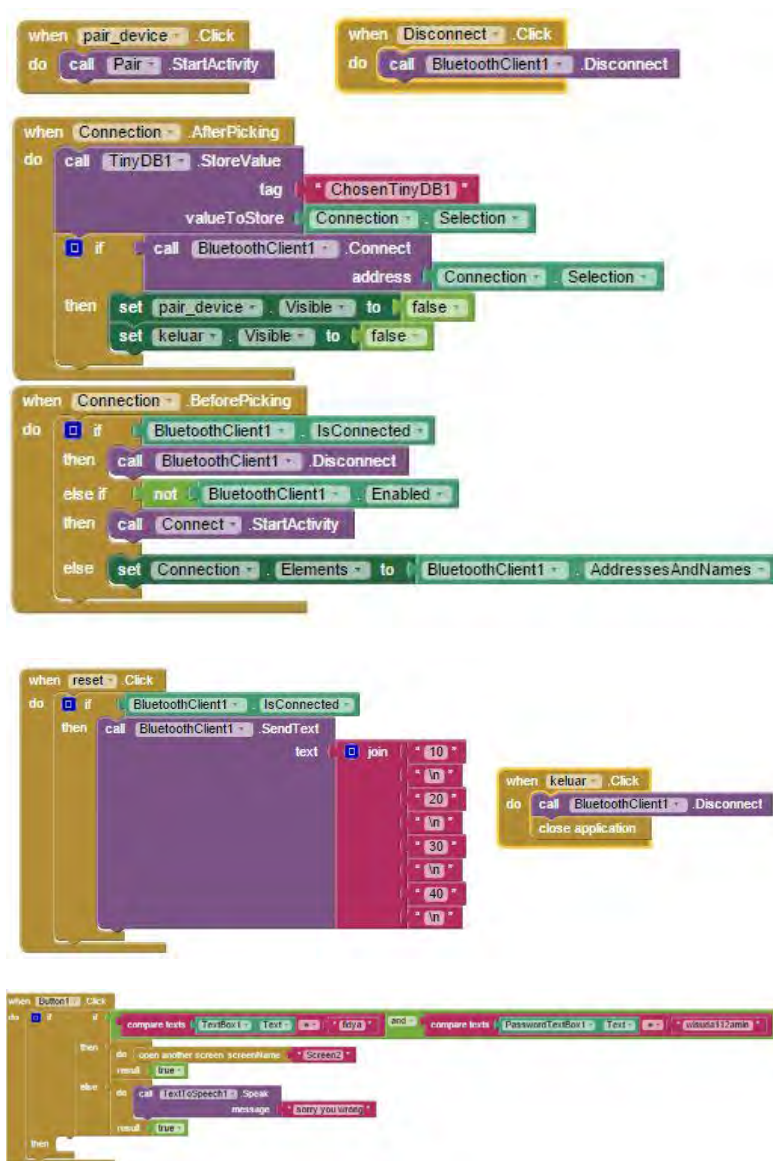
```

b. Program aplikasi di MIT App Inventor

*****/







Halaman ini sengaja dikosongkan

LAMPIRAN 2

DATA SHEET

Features

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions - Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory (ATmega48PA/88PA/168PA/328P)
 - 256/512/1K/2K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)
 - 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFNMLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFNMLF and 32-pad QFNMLF
- Operating Voltage:
 - 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P
- Temperature Range:
 - -40°C to 85°C
- Speed Grade:
 - 0 - 20 MHz @ 1.8 - 5.5V
- Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:
 - Active Mode: 0.2 mA
 - Power-down Mode: 0.1 µA
 - Power-save Mode: 0.75 µA (Including 32 kHz RTC)



8-bit AVR®
Microcontroller
with 4/8/16/32K
Bytes In-System
Programmable
Flash

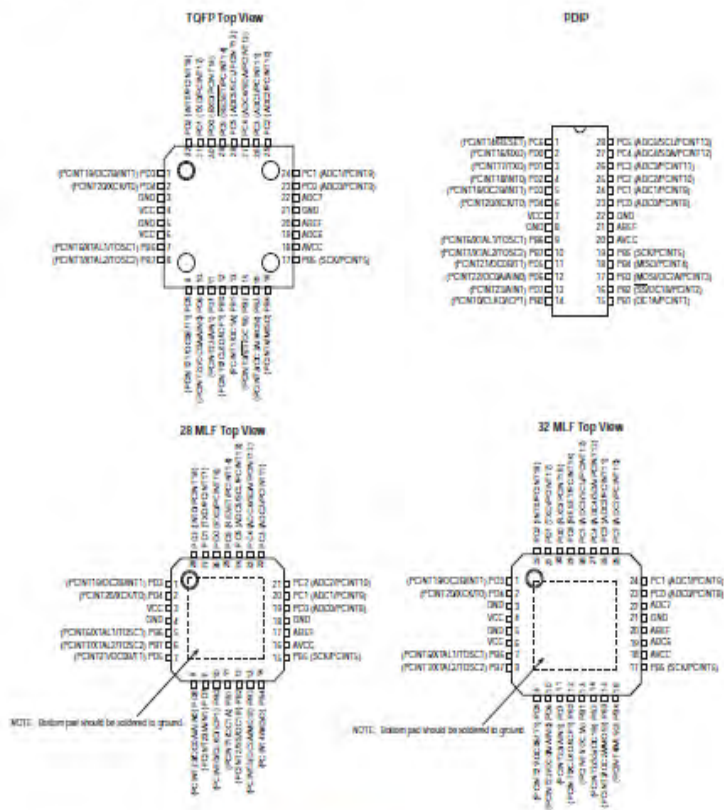
ATmega48PA
ATmega88PA
ATmega168PA
ATmega328P

Rev. 3751D-AVR-1065



1. Pin Configurations

Figure 1-1. Pinout ATmega48PA/88PA/168PA/328P





www.maxim-ic.com

DS1307 64 x 8 Serial Real-Time Clock

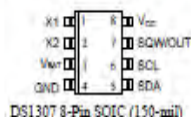
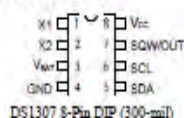
FEATURES

- Real-time clock (RTC) counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap-year compensation valid up to 2100
- 56-byte, battery-backed, nonvolatile (NV) RAM for data storage
- Two-wire serial interface
- Programmable squarewave output signal
- Automatic power-fail detect and switch circuitry
- Consumes less than 500nA in battery backup mode with oscillator running
- Optional industrial temperature range: -40°C to +85°C
- Available in 8-pin DIP or SOIC
- Underwriters Laboratory (UL) recognized

ORDERING INFORMATION

DS1307	8-Pin DIP (300-mil)
DS1307Z	8-Pin SOIC (150-mil)
DS1307N	8-Pin DIP (Industrial)
DS1307ZN	8-Pin SOIC (Industrial)

PIN ASSIGNMENT



PIN DESCRIPTION

V _{CC}	- Primary Power Supply
X1, X2	- 32.768kHz Crystal Connection
V _{BAT}	- +3V Battery Input
GND	- Ground
SDA	- Serial Data
SCL	- Serial Clock
SQW/OUT	- Square Wave/Output Driver

DESCRIPTION

The DS1307 Serial Real-Time Clock is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially via a 2-wire, bi-directional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power sense circuit that detects power failures and automatically switches to the battery supply.

Halaman ini sengaja dikosongkan

RIWAYAT HIDUP PENULIS



Nama : Fidy Eka Prahesti
TTL : Gresik, 23 Juni 1994
Jenis Kelamin : Perempuan
Agama : Islam
Alamat Rumah : JL. Kudus no 32
Gresik
Telp/HP : 089678647566
E-mail : feprahesti@gmail.com
Hobi : Berenang

RIWAYAT PENDIDIKAN

- 2000 – 2006 : SD Muhammadiyah 2 Gresik
- 2006 – 2009 : SMP Muhammadiyah 12 GKB Gresik
- 2009 – 2012 : SMA Negeri 1 Manyar Gresik
- 2012 – sekarang : Bidang Studi Komputer Kontrol, Program D3 Teknik Elektro, ITS

PENGALAMAN KERJA

- Kerja Praktek di PLTGU PJB Gresik (1 Juli – 26 Juli 2013)
- Kerja Praktek di Pt Petro Oxo Nusantara Gresik

PENGALAMAN ORGANISASI

- Staf magang Kewirausahaan HIMAD3TEKTRO 2012 – 2013
- Staf Kewirausahaan HIMAD3TEKTRO 2013 – 2014
- Staf Dana Usaha Lembaga Dakwah Jurusan 2013-2014
- Ko departemen Dana dan Kelembagaan Lembaga Dakwah Jurusan 2013-2014

Halaman ini sengaja dikosongkan

RIWAYAT HIDUP PENULIS



Nama : Khalid Oki Wibowo
TTL : Jakarta, 16 Maret 1992
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat Rumah : JL. Penggalang C No
67, Bekasi
Telp/HP : 087880555796
E-mail : khalid oki
wibowo@gmail.com
Hobi : Futsal

RIWAYAT PENDIDIKAN

- 1998 – 2004 : SDN Sepanjang Jaya VI Bekasi
- 2004 – 2007 : SMP Negeri 16 Bekasi
- 2007 – 2010 : SMA Negeri 13 Bekasi
- 2012 – sekarang : Bidang Studi Elektro Industri, Program D3 Teknik Elektro, ITS

PENGALAMAN KERJA

- Kerja Praktek di PT. PLN (Persero) DISTRIBUSI JAWA BARAT DAN BANTEN, Bekasi (27 Juni – 27 Juli 2013)
- Kerja Praktek di PTPN XI Unit Usaha PG Kanigoro, Madiun (16 Juni-16 Juli 2014)
- Karyawan Operator di PT. Astra Honda Motor (Oktober 2010-Oktober 2012)

PENGALAMAN ORGANISASI

- Staff Team Event Creative UKM ITS Badminton Community 2014 – 2015
- Staff Media dan Informasi Departement Hubungan Masyarakat LMB ITS 2014 – 2015

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2.1	Lampu Lalu Lintas	5
Gambar 2.2	Bentuk Fisik <i>Pilot Lamp</i>	6
Gambar 2.3	<i>App Inventor Designer</i>	8
Gambar 2.4	<i>Block Editor</i>	8
Gambar 2.5	Lambang Android	10
Gambar 2.6	Bluetooth HC05	12
Gambar 2.7	Mikrokontroller Arduino Uno R3	13
Gambar 2.8	Skematik Arduino Uno R3	14
Gambar 2.9	Relay 5 kaki	15
Gambar 3.1	Diagram Alat Secara Keseluruhan	17
Gambar 3.2	Layout jalan simpang empat	19
Gambar 3.3	Kotak Penyimpanan Alat Elektronik	19
Gambar 3.4	Kotak Akrilik Lampu Lalu Lintas	20
Gambar 3.5	Tiang Penyangga Lampu Lalu Lintas	20
Gambar 3.6	Skematik Rangkaian Power Supply 9V	22
Gambar 3.7	Skematik Rangkaian Power Supply 5V	22
Gambar 3.8	Bentuk Fisik Rangkaian <i>Power Supply</i> 9V dan <i>Power Supply</i> 5V	23
Gambar 3.9	Konfigurasi Pin ATmega 328/Arduino Uno	23
Gambar 3.10	Skematik Mikrokontroler	24
Gambar 3.11	Skematik Relay	25
Gambar 3.12	Bentuk Fisik <i>Relay Driver</i>	25
Gambar 3.13	Komunikasi Serial Bluetooth dan Arduino	26
Gambar 3.14	Skematik Rangkaian Serial Bluetooth HC-05 dan Arduino Uno R3	27
Gambar 3.15	Skematik RTC	28
Gambar 3.16	Gambar Fisik RTC	28
Gambar 3.17	Diagram Blok Penambahan Time Delay Lampu Lalu Lintas	29
Gambar 3.18	Serial Monitor Pada Tampilan Program	30
Gambar 3.19	Tampilan Awal MIT App Inventor 2	32
Gambar 3.20	Tampilan App Designer	33
Gambar 3.21	<i>Layout</i> Aplikasi	34
Gambar 3.22	Tampilan Desain Blok Awal	34
Gambar 3.23	Logic block Pengaturan Delay Time Traffic Light	35
Gambar 3.24	Flowchart Program MIT App Inventor	36
Gambar 4.1	Implementasi Rangkaian Kontrol Pengatur	

	<i>Traffic light</i>	37
Gambar 4.2	Pengujian <i>Input</i> dan <i>Output</i> Mikrokontroler Arduino Uno	39
Gambar 4.3	Pengujian Rangkaian <i>Power Suply</i> 5V di Project Board	42
Gambar 4.4	Pengujian Rangkaian <i>Power Suply</i> 9V di Project Board	42
Gambar 4.5	(a) Mengoneksikan Bluetooth Dengan PC dan Arduino (b) Monitoring Data Bluetooth	45
Gambar 4.6	Pengujian RTC	46
Gambar 4.7	<i>Serial Monitor</i> menjalankan <i>Traffic light</i>	47
Gambar 4.8	Pemberian Waktu <i>Delay</i> pada lampu hijau 1	47
Gambar 4.9	Pemberian Waktu <i>Delay</i> pada lampu hijau 2	48
Gambar 4.10	Pemberian Waktu <i>Delay</i> pada lampu hijau 3	48
Gambar 4.11	Pemberian Waktu <i>Delay</i> pada lampu hijau 4	48
Gambar 4.12	Proses Reset Progam Ke Kondisi Semula	49
Gambar 4.13	Perancangan Lay Out Lampu Lalu Lintas	52



SIDANG TUGAS AKHIR

Rancang Bangun Pengaturan Lampu Lalu lintas Berbasis Mikrokontroler yang dikendalikan dengan aplikasi di Android secara Nirkabel

Fidya Eka Prahesti
Khalid Oki Wibowo

NRP. 2212 039 010
NRP. 2212 039 033

Dosen Pembimbing I :
Suwito ,S.T,M.T
NIP: 19810105 20050 1 004

Dosen Pembimbing II :
Eko Pujiyatno Matni, S.Pd
NIP:19710330 199403 1 002

Latar Belakang



POLISI HARUS MENGATUR LALU LINTAS DENGAN ISYARAT TANGAN

Latar Belakang

International Data Corporation (IDC) 2015



Worldwide Smartphone Operating System 2011 and 2015 Market Share and 2011-2015 Compound Annual Growth Rate

Operating System	2011 Market Share	2015 Market Share	2011-2015 Unit CAGR
Android	38.9%	43.8%	23.7%
BlackBerry OS	14.2%	13.4%	18.3%
Symbian	20.6%	0.1%	-68.8%
iOS	18.2%	16.9%	17.9%
Windows Phone 7/Windows Mobile	3.8%	20.3%	82.3%
Others	4.3%	5.5%	27.6%
Total	100.0%	100.0%	20.1%

<http://www.indopost.com/blog/2011/06/worlds-smartphone-operating-system-market-share-in-2011-39-google-android-apple-ios-18-rim-blackberr.html>

Permasalahan

1. Membuat sistem kontrol lampu lalu lintas secara nirkabel.
2. Menampilkan *interface* pengaturan nyala lampu menggunakan aplikasi di android.
3. Mensinkronisasikan kerja antara aplikasi di android, bluetooth dan progam di arduino .



Maksud dan Tujuan



1. Membuat konektifitas antara aplikasi di android dengan arduino.

2. Membuat sistem kontrol *traffic light* yang pengoperasiannya berdasarkan waktu yang dapat diatur oleh polisi lalu lintas.

Maksud dan Tujuan



3. Memberikan kemudahan pada polisi lalu lintas untuk mengatur kepadatan pada ruas jalan di *traffic light*

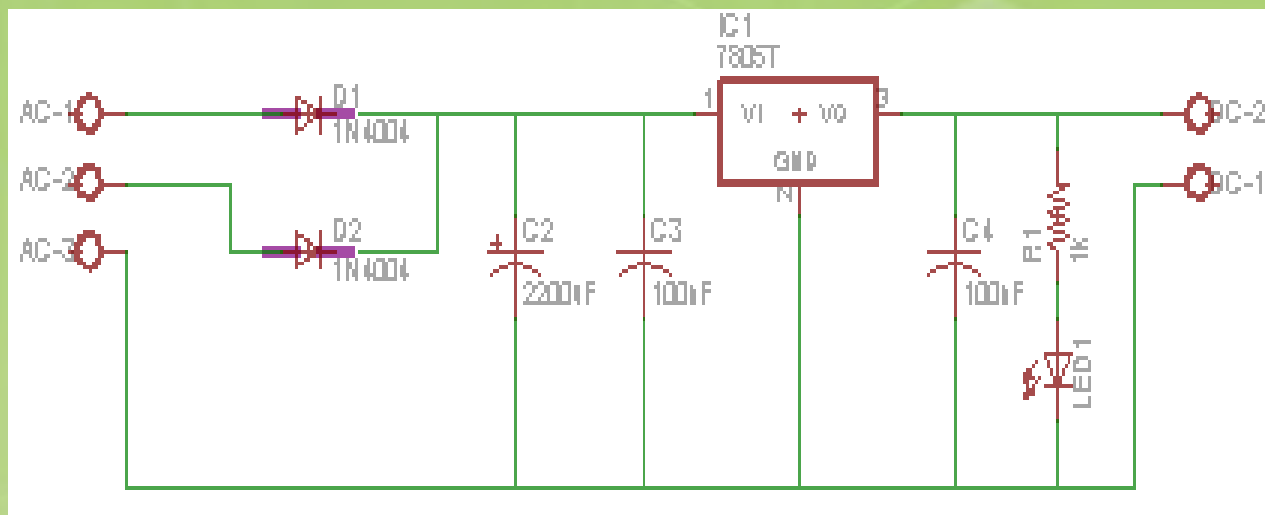
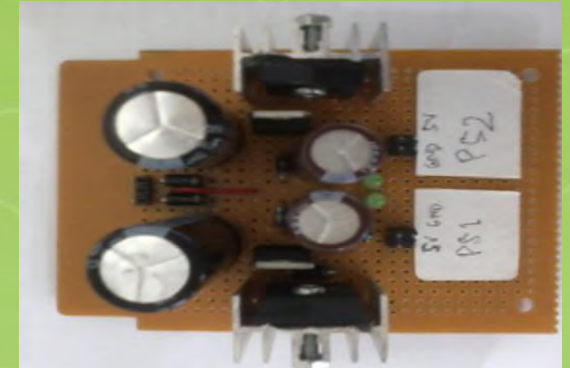
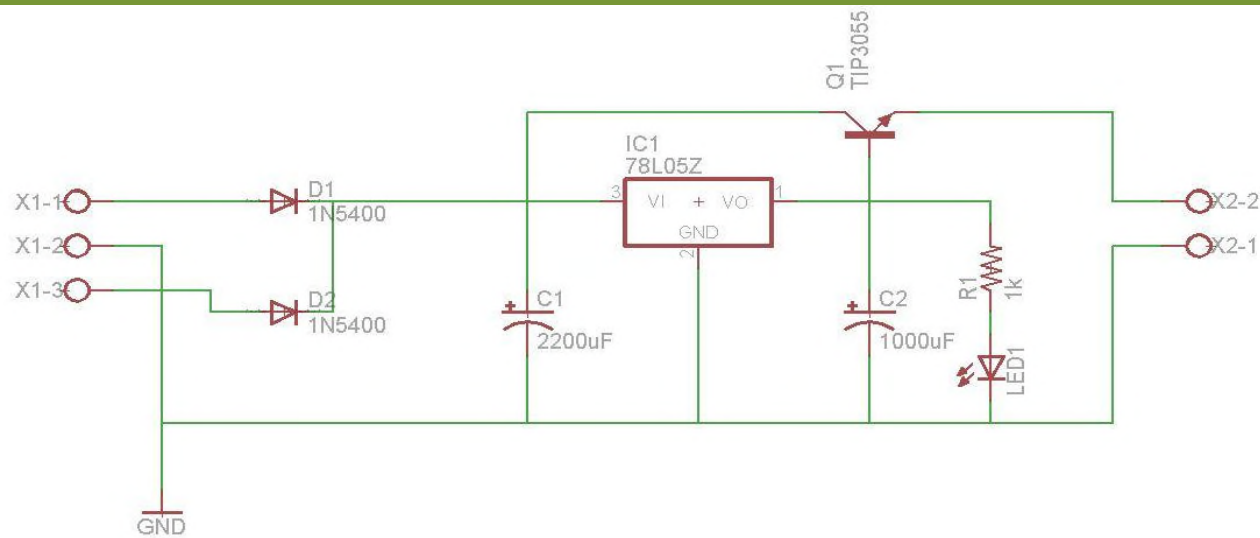
Batasan Masalah

1. Jumlah ruas jalan yang digunakan untuk dikontrol hanyalah persimpangan empat.
2. Pengujian sistem *traffic light* yang ditampilkan di aplikasi android hanya penambahan *delay* lampu hijau.
3. Koneksi antara arduino dengan aplikasi di android menggunakan komunikasi bluetooth HC-05.
4. Dibutuhkan operator atau polisi lalu lintas untuk mengatur kepadatan pada traffig light



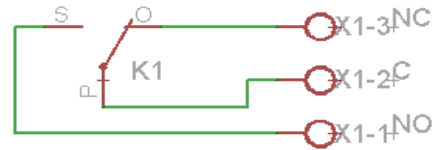
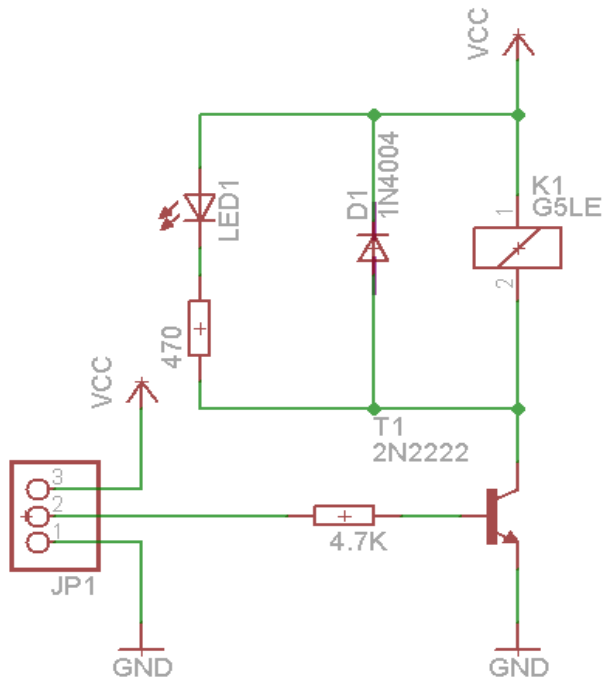
PERANCANGAN *HARDWARE*

① POWER SUPPLY



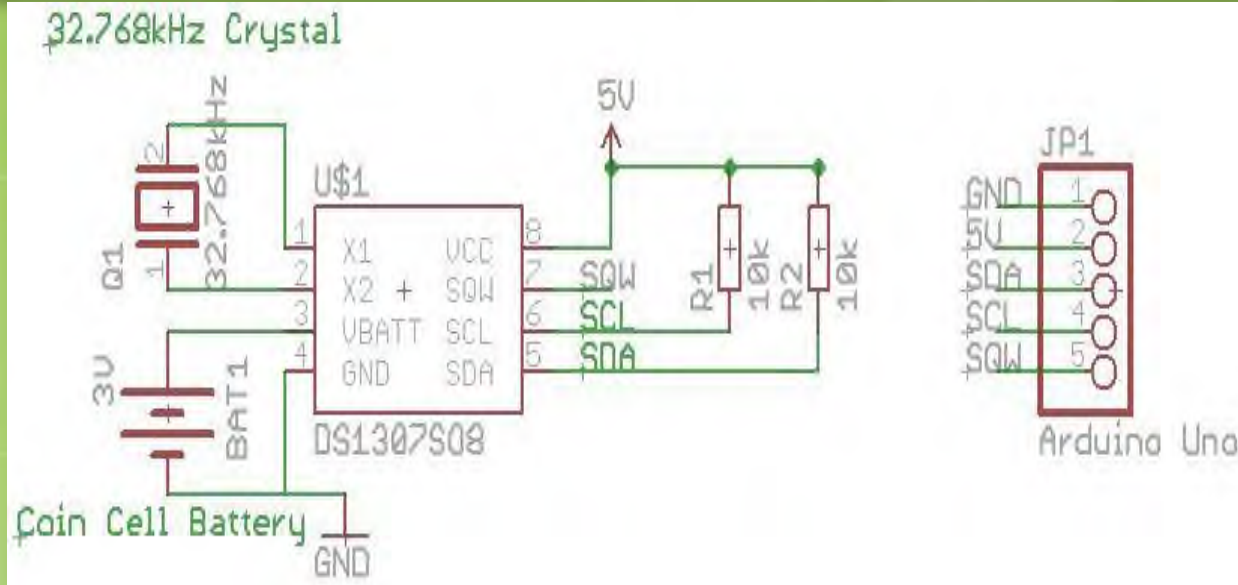
PERANCANGAN *HARDWARE*

② DRIVER RELAY



PERANCANGAN *HARDWARE*

③ RTC DS1307



④ PROTOTYPE TRAFFIC LIGHT

PERANCANGAN SOFTWARE



1

APP Inventor designer

APP Inventor Bloks Editor

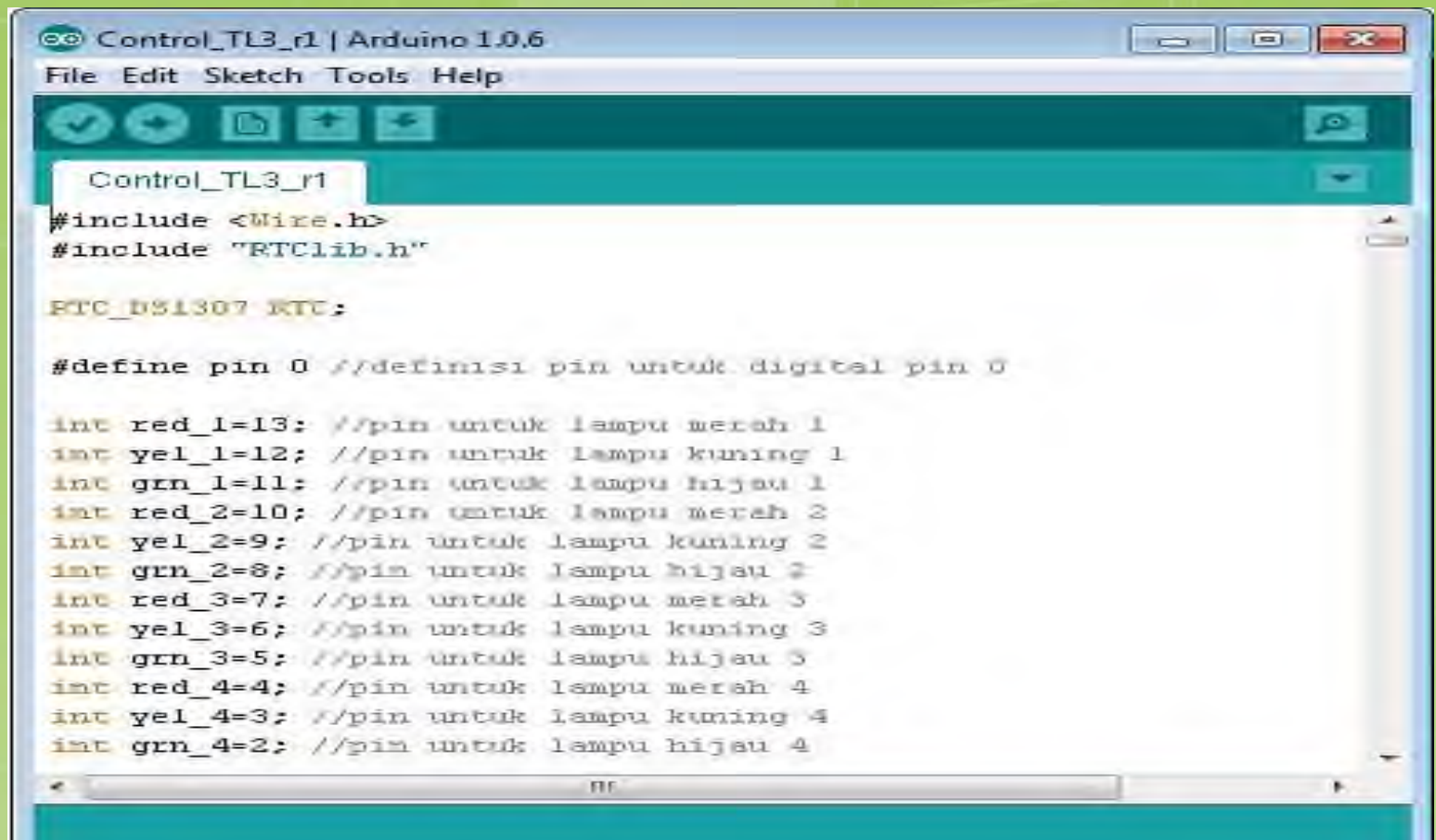
2



PERANCANGAN SOFTWARE

③

Progam Mikrokontroler



The screenshot shows the Arduino IDE interface with a sketch named "Control_TL3_r1". The code is written in C++ and includes the following components:

- Includes: `<Wire.h>` and `"RTClib.h"`.
- Library: `RTC_DS1307 RTC;`
- Pin Definition: `#define pin 0 //definisi pin untuk digital pin 0`
- Pin Assignments: A series of `int` declarations for 16 different pins, grouped by color and number (1-4).
 - Red: `red_1=13`, `red_2=10`, `red_3=7`, `red_4=4`
 - Yellow: `yel_1=12`, `yel_2=9`, `yel_3=6`, `yel_4=3`
 - Green: `grn_1=11`, `grn_2=8`, `grn_3=5`, `grn_4=2`

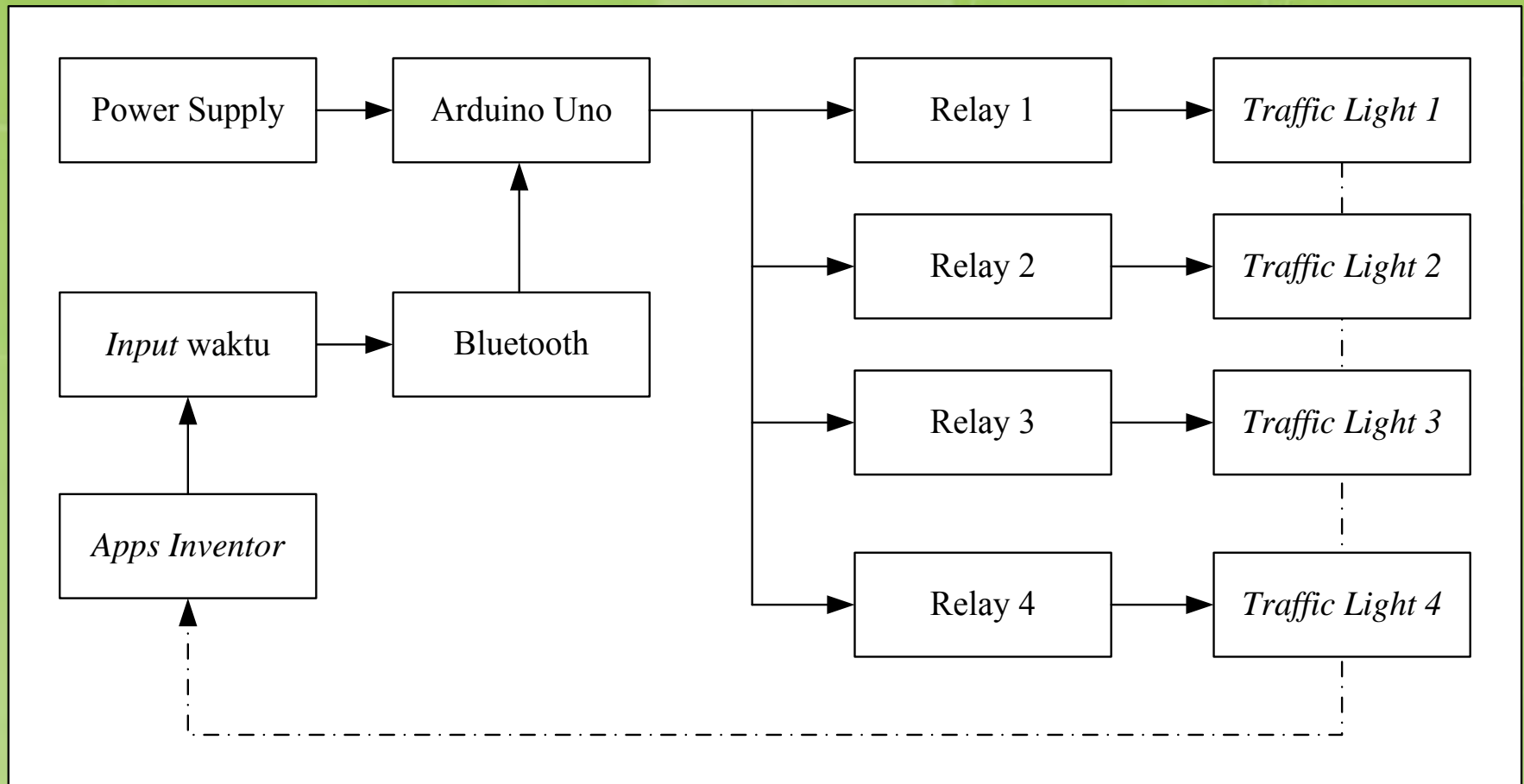
```
#include <Wire.h>
#include "RTClib.h"

RTC_DS1307 RTC;

#define pin 0 //definisi pin untuk digital pin 0

int red_1=13; //pin untuk lampu merah 1
int yel_1=12; //pin untuk lampu kuning 1
int grn_1=11; //pin untuk lampu hijau 1
int red_2=10; //pin untuk lampu merah 2
int yel_2=9; //pin untuk lampu kuning 2
int grn_2=8; //pin untuk lampu hijau 2
int red_3=7; //pin untuk lampu merah 3
int yel_3=6; //pin untuk lampu kuning 3
int grn_3=5; //pin untuk lampu hijau 3
int red_4=4; //pin untuk lampu merah 4
int yel_4=3; //pin untuk lampu kuning 4
int grn_4=2; //pin untuk lampu hijau 4
```

DIAGRAM BLOK SISTEM



Pengujian jarak koneksi *Bluetooth*

	PENGUJIAN KE				
JARAK (METER)	1	2	3	4	5
2	CONNECT	CONNECT	CONNECT	CONNECT	CONNECT
4	CONNECT	CONNECT	CONNECT	CONNECT	CONNECT
6	CONNECT	CONNECT	CONNECT	CONNECT	CONNECT
8	CONNECT	CONNECT	CONNECT	CONNECT	CONNECT
10	CONNECT	CONNECT	CONNECT	CONNECT	CONNECT
12	CONNECT	CONNECT	CONNECT	CONNECT	CONNECT
14	DISCONNECT	DISCONNECT	DISCONNECT	DISCONNECT	DISCONNECT

Tujuan dari pengambilan jarak koneksi ini adalah untuk mengetahui seberapa jauh jarak bluetooth yang kami gunakan dapat terkoneksi atau tidak. Dari hasil pengujian dapat kami analisa setelah jarak $\pm 13\text{m}$ bluetooth tidak dapat terkoneksi.

Pengujian RTC

PENGUJIAN KE	HARI 1		HARI 2	
	JAM DIGITAL LAPTOP	RTC	JAM DIGITAL LAPTOP	RTC
1	12:25:10	12:25:05	12:25:23	12:25:13
2	12:45:27	12:45:22	12:45:19	12:45:09
3	13:05:14	13:05:09	13:05:15	13:05:05
4	13:25:33	13:25:28	13:25:47	13:25:37
5	13:40:21	13:40:16	13:40:58	13:40:48
6	13:50:07	13:50:02	13:50:23	13:50:13
7	14:00:18	14:00:13	14:00:36	14:00:26
8	14:10:41	14:10:36	14:10:44	14:10:34
9	14:20:19	14:20:14	14:20:47	14:20:37
11	14:40:44	14:40:39	14:40:37	14:40:27
12	14:50:51	14:50:46	14:50:18	14:50:08
13	15:00:27	15:00:22	15:00:11	15:00:01

Hasil Pengujian

- JIKA ARDUINO PADA LOGIKA 0 MAKA *OUTPUTNYA* BERNILAI 0V
- LOGIKA 1 MAKA *OUTPUTNYA* BERNILAI 4,98V
- JARAK UNTUK MENGIRIM DAN MENERIMA DATA DARI BLUETOOTH SEJAUH 1-12 METER.
- TEGANGAN KELUARAN RATA-RATA PADA *POWER SUPPLY* 5VOLT TANPA BEBAN ADALAH 4,99V DENGAN PERSENTASE KESALAHAN PADA RANGKAIAN SEBESAR 0,2 %
- TEGANGAN KELUARAN RATA-RATA PADA *POWER SUPPLY* 9VOLT DENGAN BEBAN ADALAH 9 V DENGAN PERSENTASE KESALAHAN PADA RANGKAIAN SEBESAR 0 %
- RATA-RATA KERJA *RELAY* YANG KAMI UJI SESUAI DENGAN OUTPUT NYALA LAMPU PADA LAY OUT *TRAFFIC LIGHT*.
- PADA SAAT PENGUJIAN PROGAM APP INVENTOR DENGAN *DELAY* 10 DETIK DI LAMPU HIJAU PERTAMA TINGKAT KEBERHASILAN : 0% GAGAL DAN 100% BERHASIL.

KESIMPULAN

- Jarak komunikasi *Bluetooth* yang dapat dilakukan untuk mengirim dan menerima data adalah sejauh 12 ± 1 meter.
- Pergeseran waktu antara jam RTC dengan jam digital laptop antara 5-10 detik. Sehingga RTC dapat menjadi acuan untuk time shedule lampu lalu lintas.
- Error yang dihasil dari tegangan keluaran Power Supply 5volt sebesar 0,10 - 0,12 volt dan pada power supply 9 volt sebesar 0 volt.
- Penambahan delay waktu dengan app inventor baik satuan maupun puluhan dapat diterima dengan baik dengan presentase error sebesar 0%





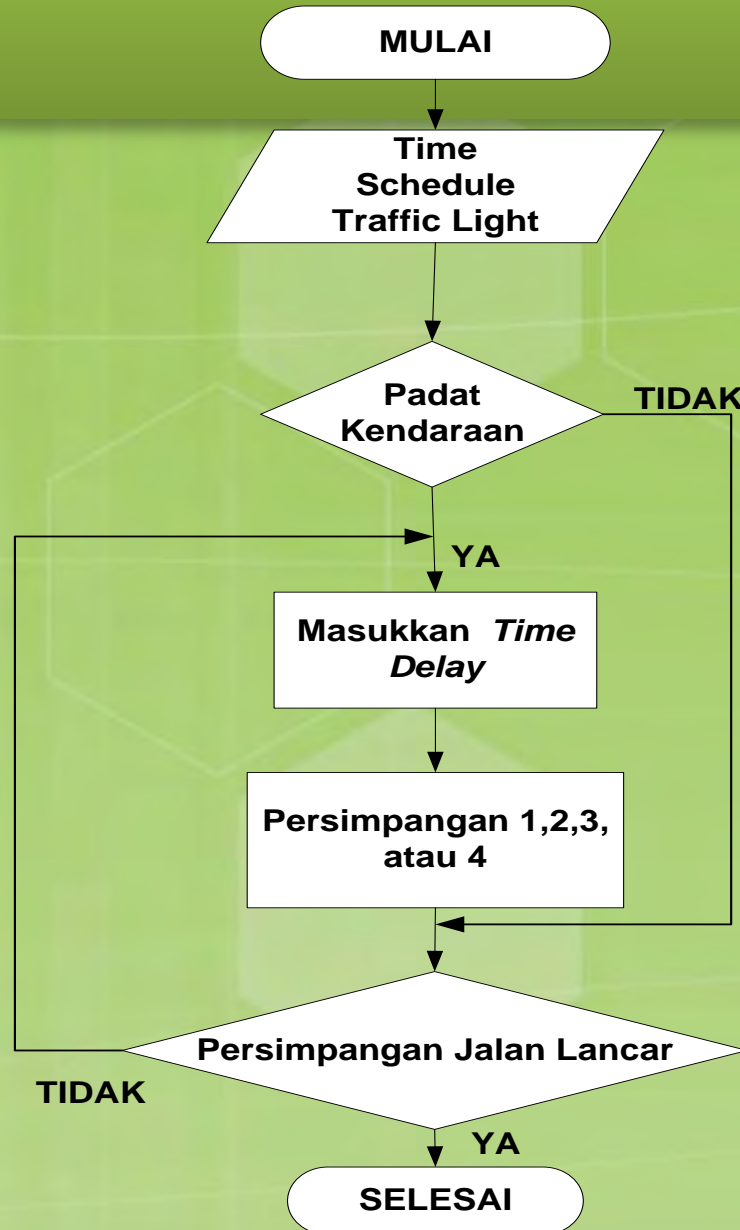
😊 TERIMAKASIH 😊



Cara Kerja Alat



FLOWCHART TRAFFIC LiGHT



Flow chart MIT APP Inventor

